



Instrumentation  
Technologies

**Libera Workshop**

October 2006

1 / 20

# **CSPI: Integrating Libera into Control System(s)**

**Aleš Bardorfer**

**Instrumentation Technologies  
Slovenia**

[ales@i-tech.si](mailto:ales@i-tech.si) ; [www.i-tech.si](http://www.i-tech.si)

**Aleš Bardorfer**



# 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**

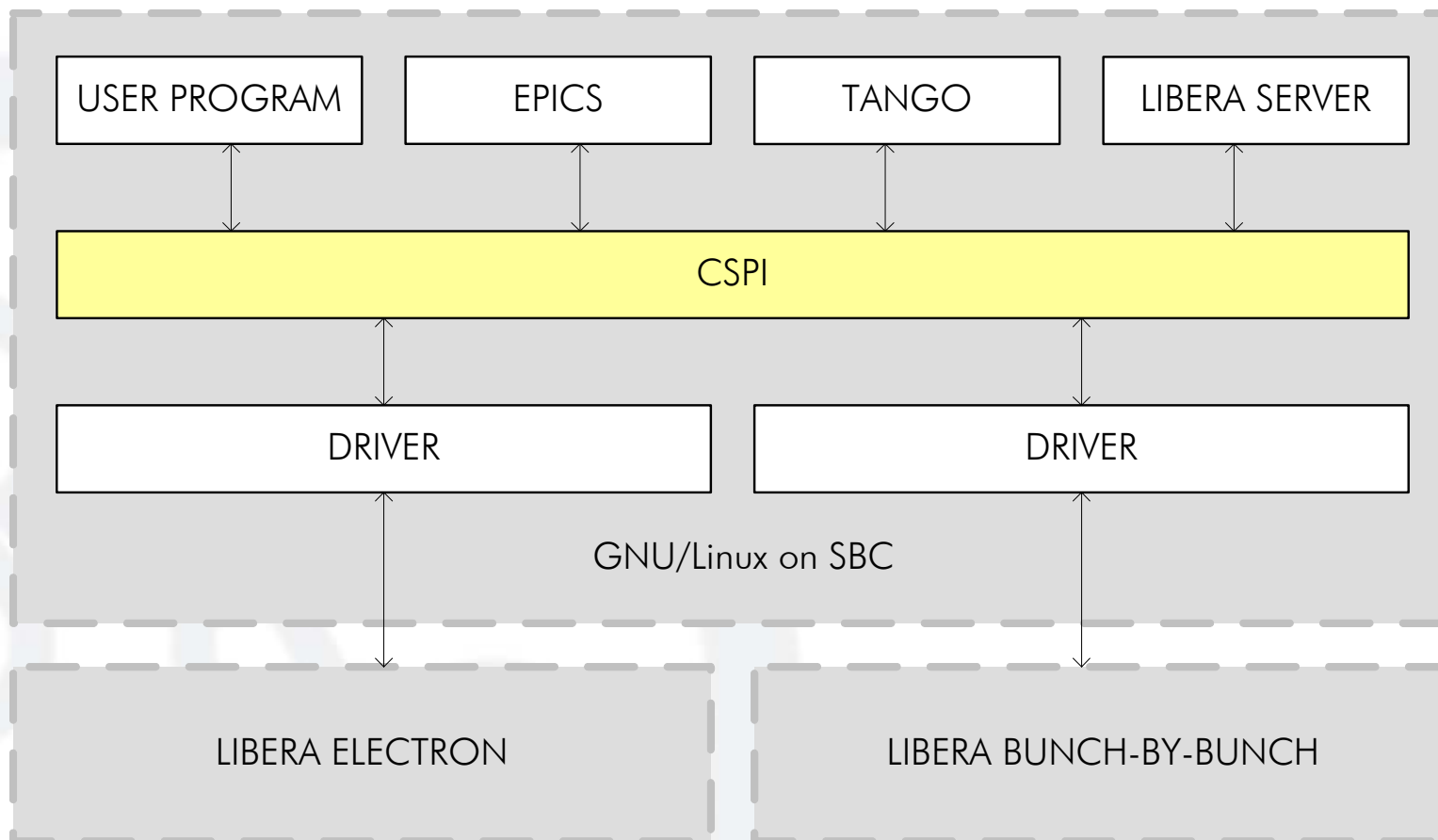


## What is CSPI?

- **CSPI – Control System Programming Interface**
- **High-level C interface**
- **Library to be linked with user application**
- **Standard Libera API**
- **User's shield for proper Libera use:**
  - **Longterm API compatibility**
  - **Prevents from Libera use scenarios that don't make any sense**
  - **Hides HW details and extracts data paths**



# CSPI Structure





## **CSPI and Libera Family**

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

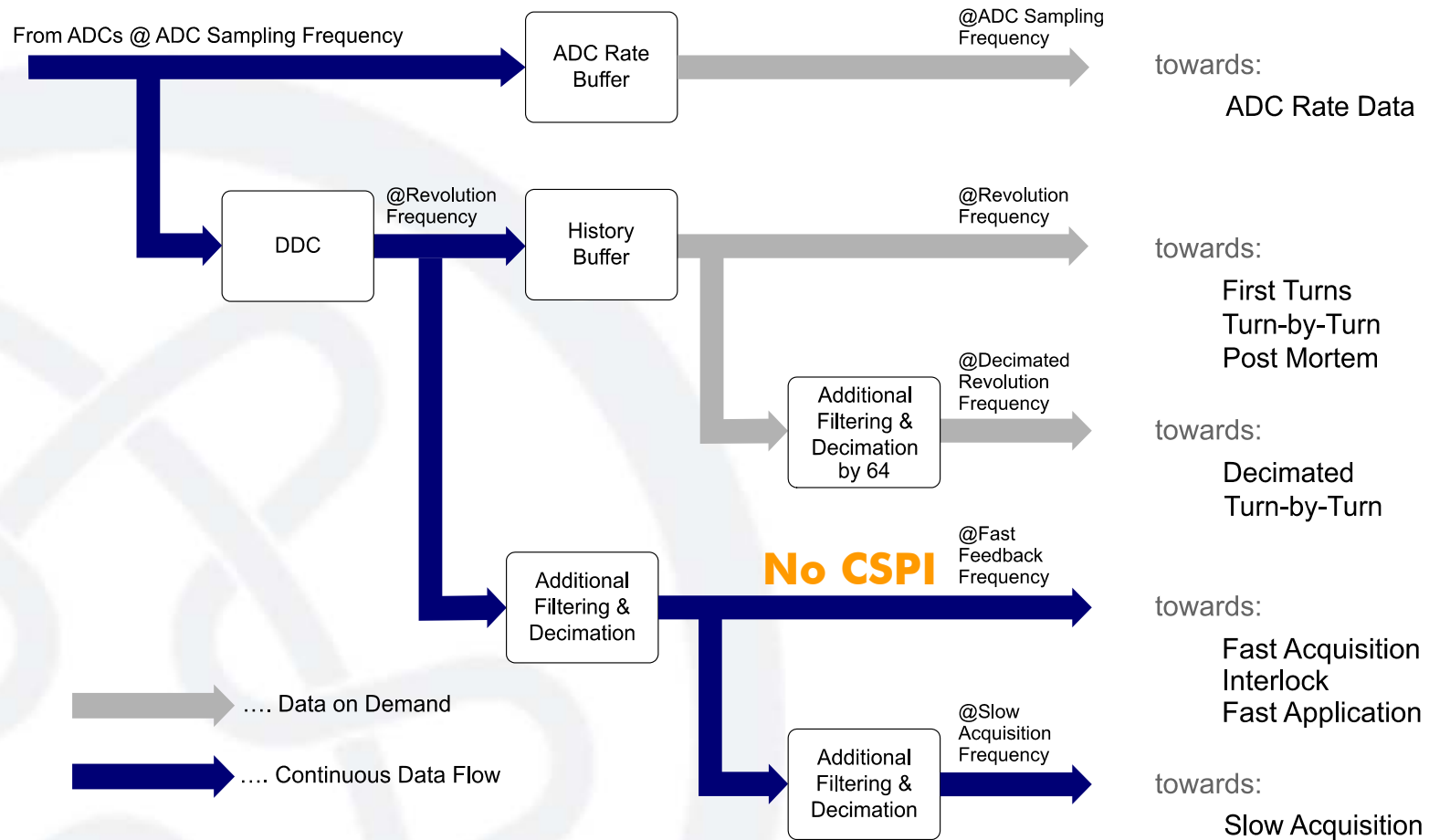


## **CSPI Roles**

- **Libera as a standard API**
  - **Relatively constant API**
  - **Covers a whole range of processors**
- **A Shield for the user:**
  - **“Proper” use of Libera**
  - **Multi-user safe**
  - **Multi-application safe**



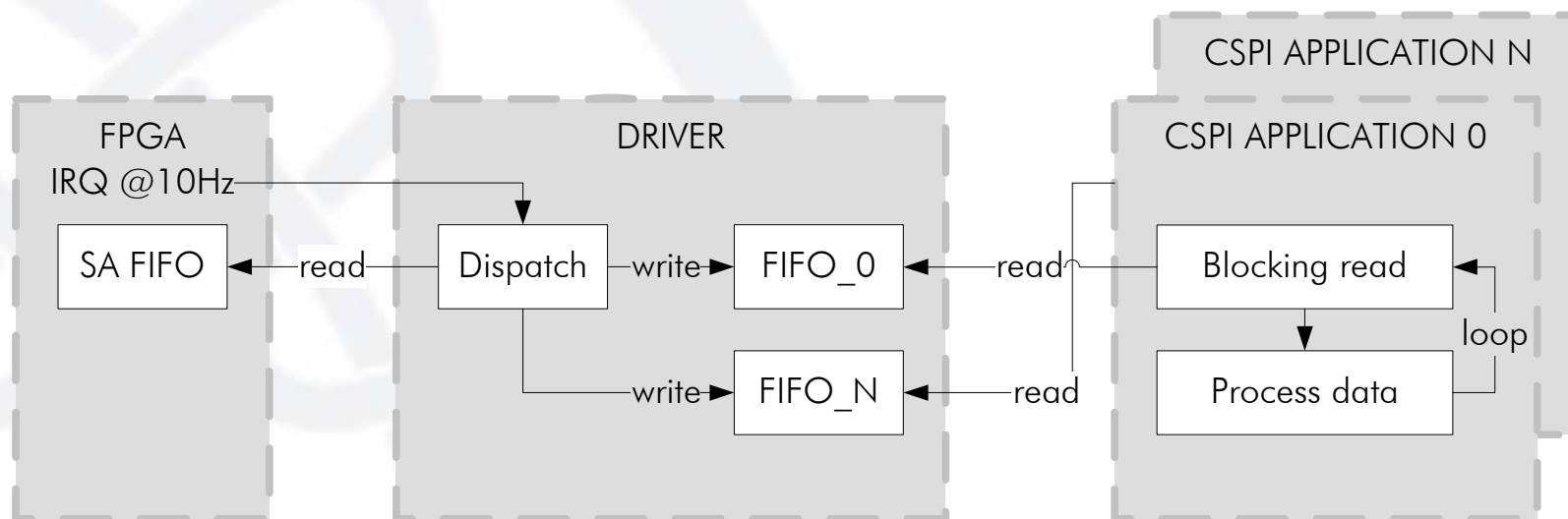
# Libera Data Sources





# Streaming Data Paths

- **Slow acquisition (SA)**
- **Data delivery frequency dictated by FPGA (synchronous to TbT)**
- **cspi\_get()**





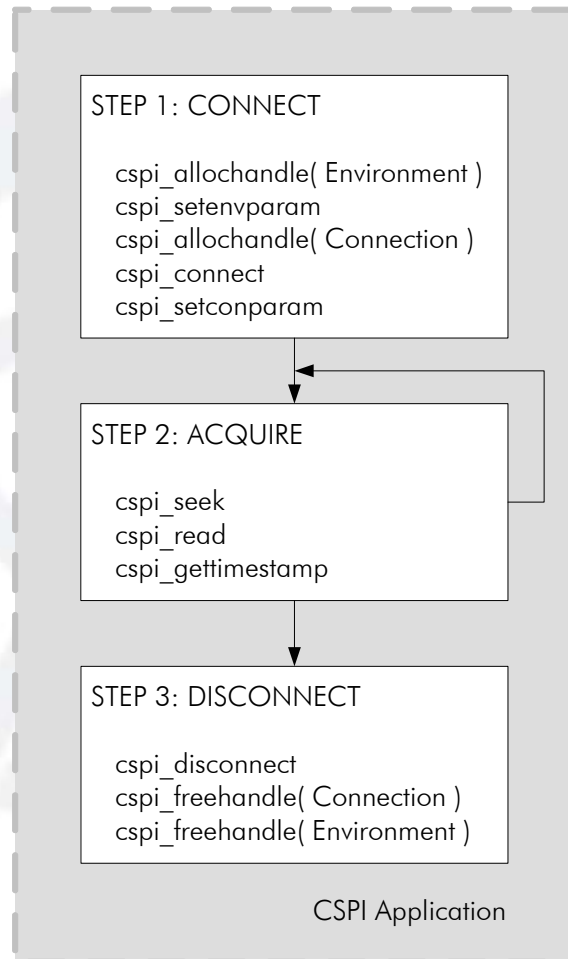


## **Non-Streaming Data Paths**

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



## General CSPI Application Flow

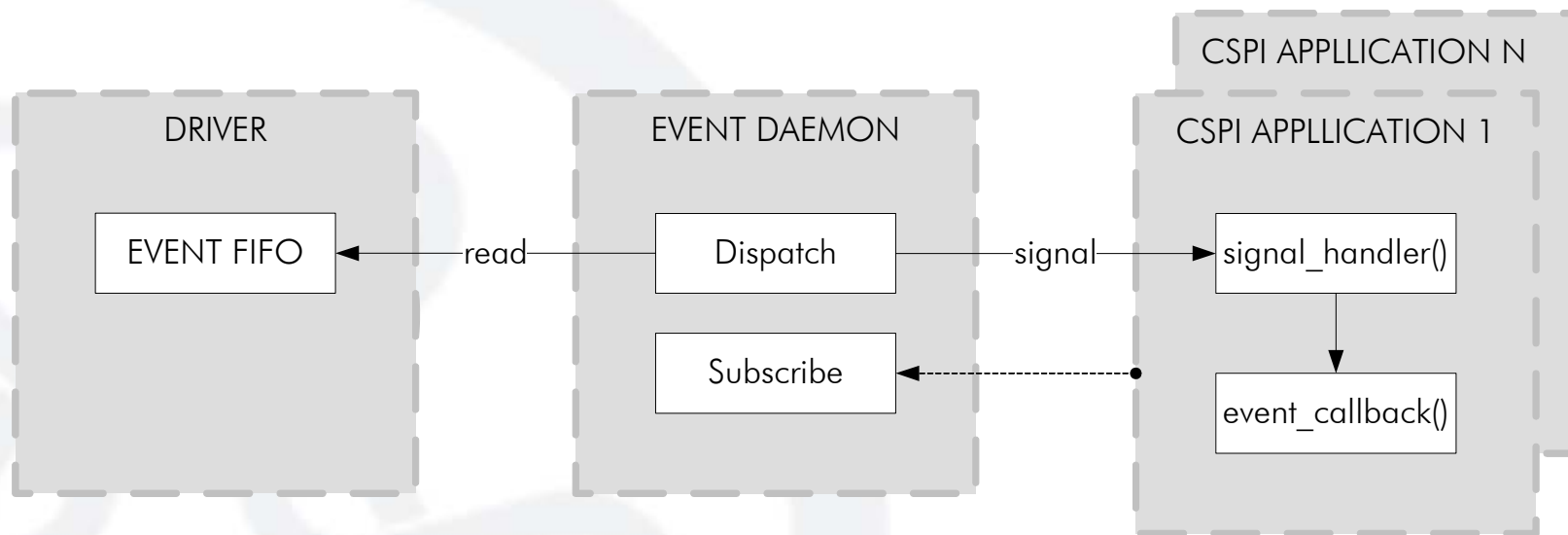


- **Handles:**
  - **Environment**
  - **Connection**
- **Parameters**
- **Data retrieval functions:**
  - **`cspi_seek()`**
  - **`cspi_read()`**
  - **`cspi_get*()`**



# Async Notification Mechanism

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





## Discussion

- **SA:**
  - **Blocking read()**
  - **Non-blocking read()**
- **Event notification:**
  - **Linux signals**
  - **Message pumps through FIFOs**
- **Set\_time() units:**
  - **MT vs. LMT**



## Slow Acquisition (SA), R-1.40

- **Blocking read()**
- **Non-blocking read():**
  - Returns a system error (**EAGAIN**) if no data is available
  - **Select()** call
  - **ioctl(FIONREAD)** possible
  - ...?
- **CSPI:**
  - New SA specific connection parameter and a corresponding connection flag (**CSPI\_CON\_SANONBLOCK**)



# Integration into Control Systems

- **The Problem:**
  - 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



# EPICS

- **Current state - two EPICS drivers:**
  - **DLS driver:**
    - lots of additions to basic libera functionality
    - DLS specifics
  - **Instrumentation Technologies driver:**
    - Basic, lightweight
    - Resembles basic Libera data paths
    - Based on ASYN EPICS driver
- **Both use CSPI as a low level SW layer**
- **Possible merging:**
  - **Get the best features out of two drivers**



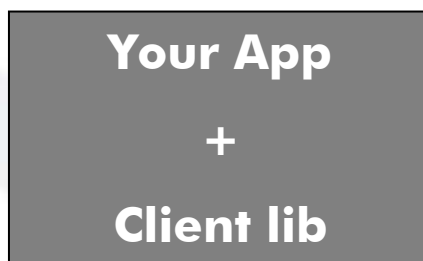
# Tango

- **First generation:**
  - i386 based
  - uses Libera generic server and client library
- **Second generation:**
  - Run on Libera (ARM)
  - Ported to Libera by Soleil, Elettra, ...



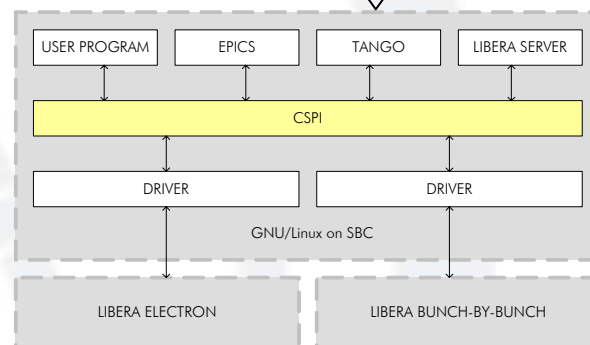
# Generic Server & Client Library

**Powerful PC,  
Workstation**



**CSPI API +  
server\_connect()  
server\_disconnect()**

**Libera (ARM)**



**CSPI API**



# Generic Server & Client Library

- **Reflects the CSPI API to the client side**
- **Network transparent CSPI API**
- **Runs on any UNIX based OS**
- **Implements additional cache on the server side to avoid network congestion problems**
- **Current and future use:**
  - **Tango**
  - **Matlab**



# Integration into Other Control Systems

- **On Libera (ARM):**
  - **Development toolchain available**
  - **Linux 2.4, 2.6 coming soon**
  - **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**
- **GPL – Full open source code eases integration**