LLRF and Diagnostics Architecture

Kay Rehlich
DESY
9. 2. 2006

Proposal for a common architecture

Version 0

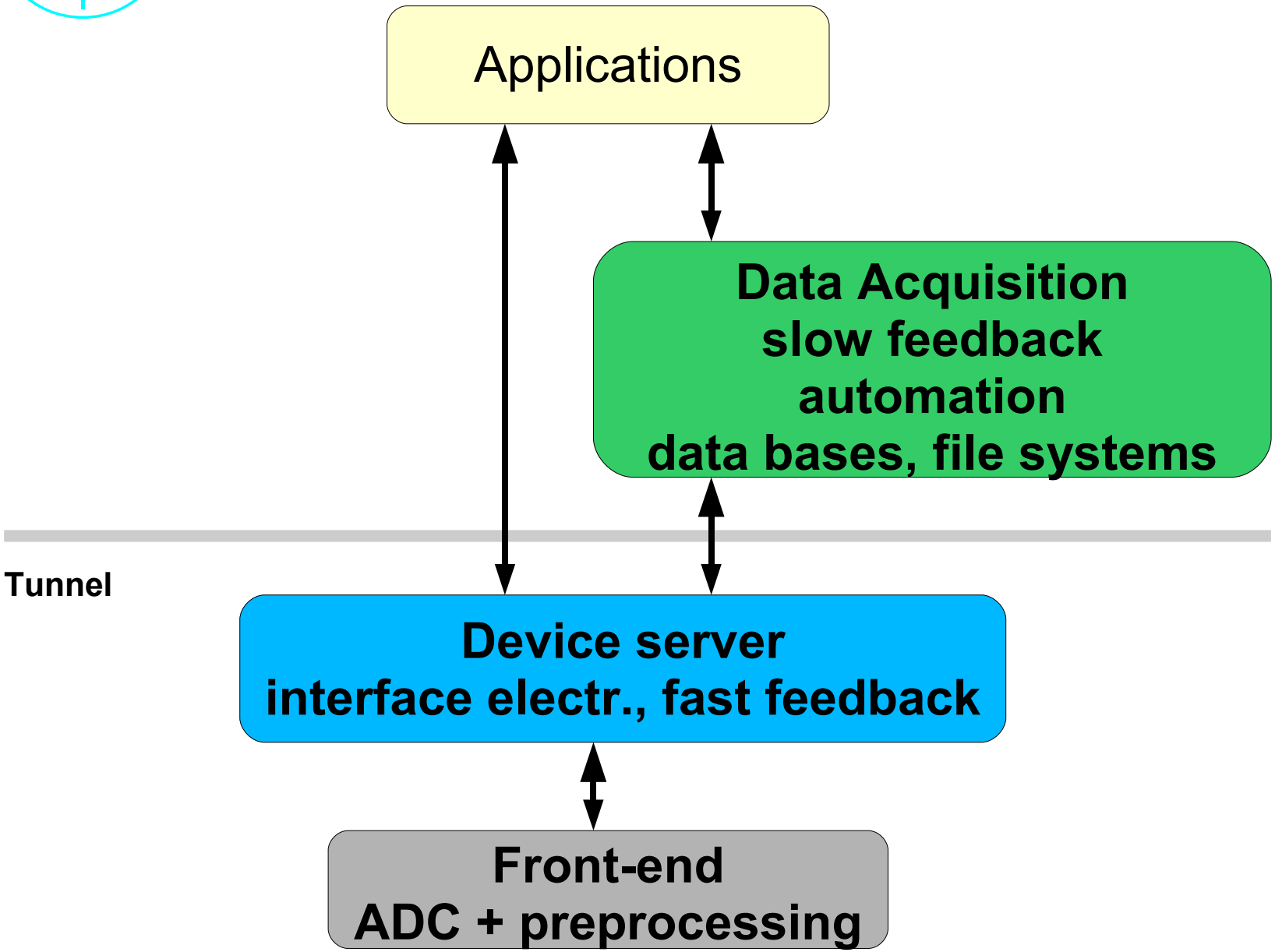
Motivation

- **Common developments for the XFEL and ILC and ...?**
- **Can we define/develop common hardware and interfaces ?**

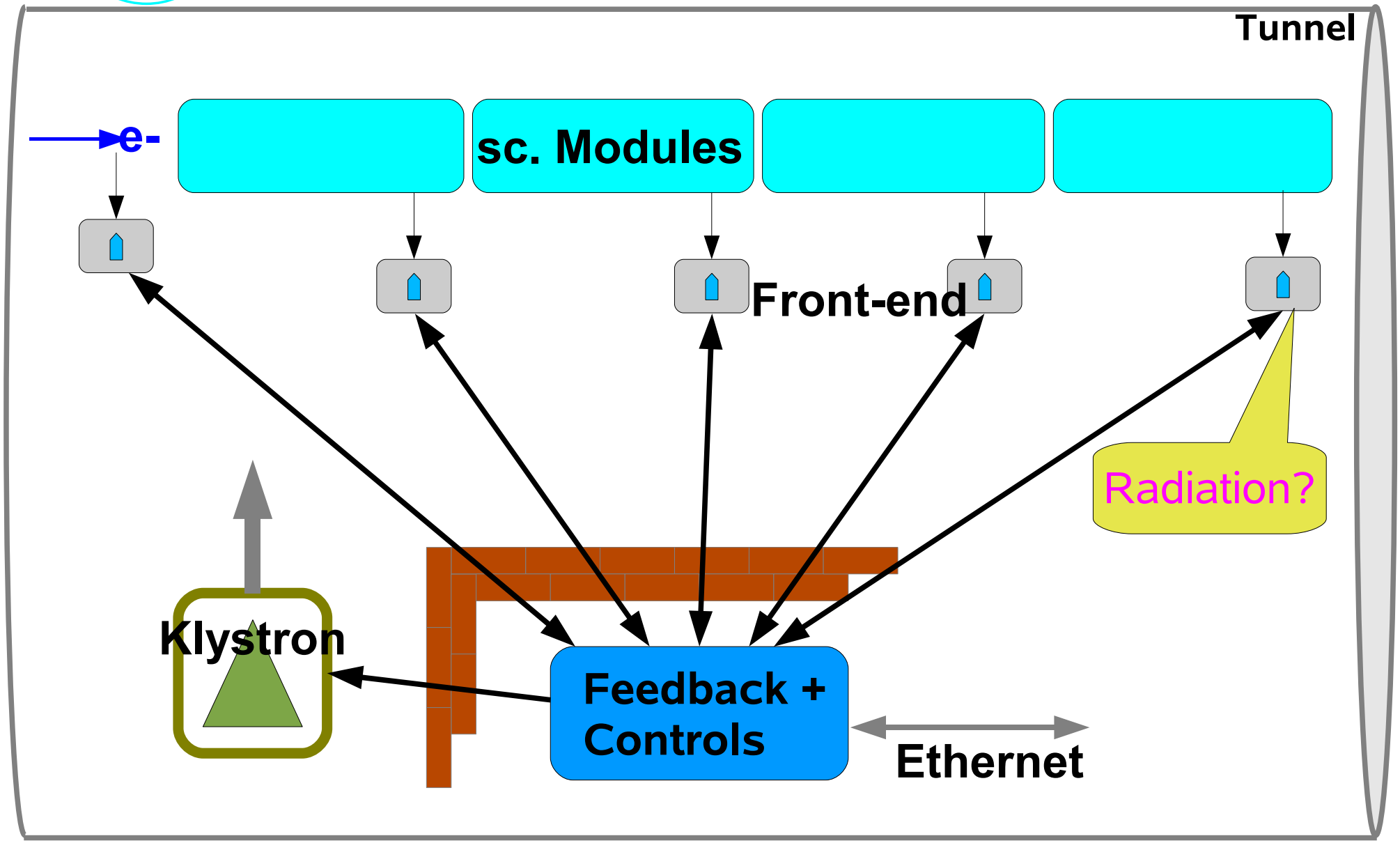
Goals

- Define hardware protocols for:
  - Feedback data transfers
  - Timing
  - Data transfer to the control system
  - Parameter transfer
  - Program loading
  - Debug access
- Define software interfaces
  - Where to run algorithms and processes

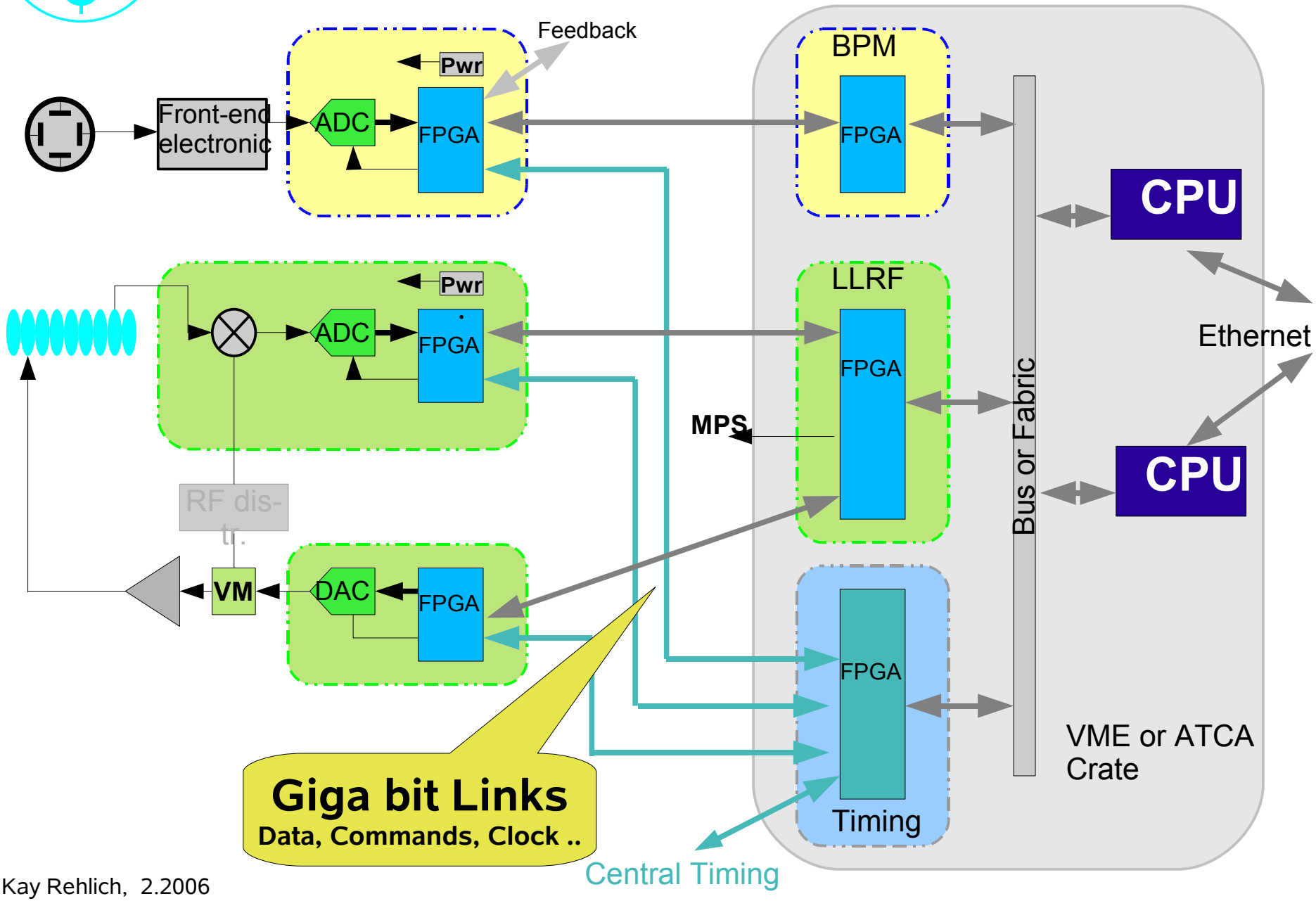
Control System Blocks



Geographical Layout

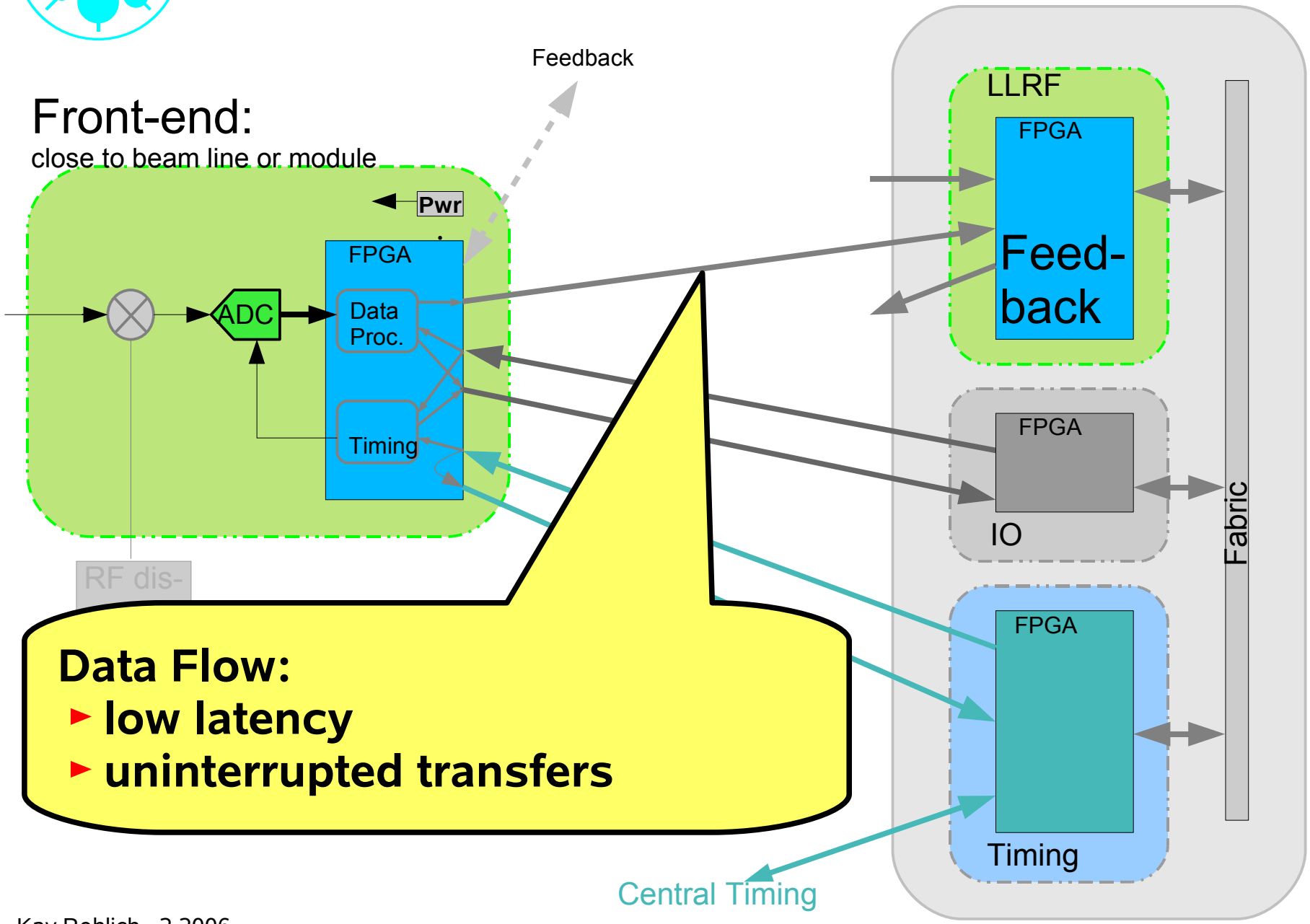


RF and Diag Building Blocks



Kay Rehlich, 2.2006

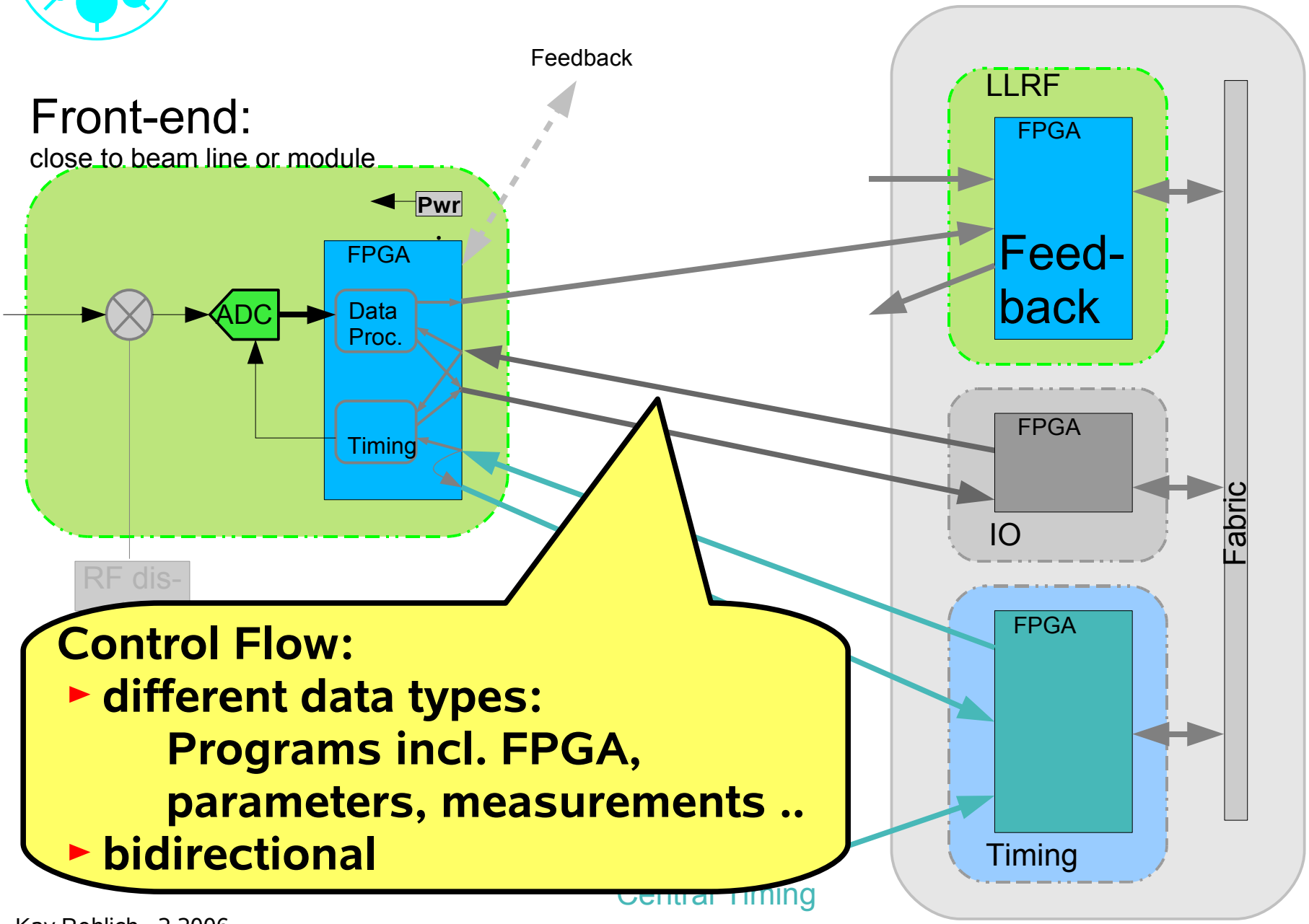
Gbit Links: Feedback Data



Data Flow:

- **low latency**
- **uninterrupted transfers**

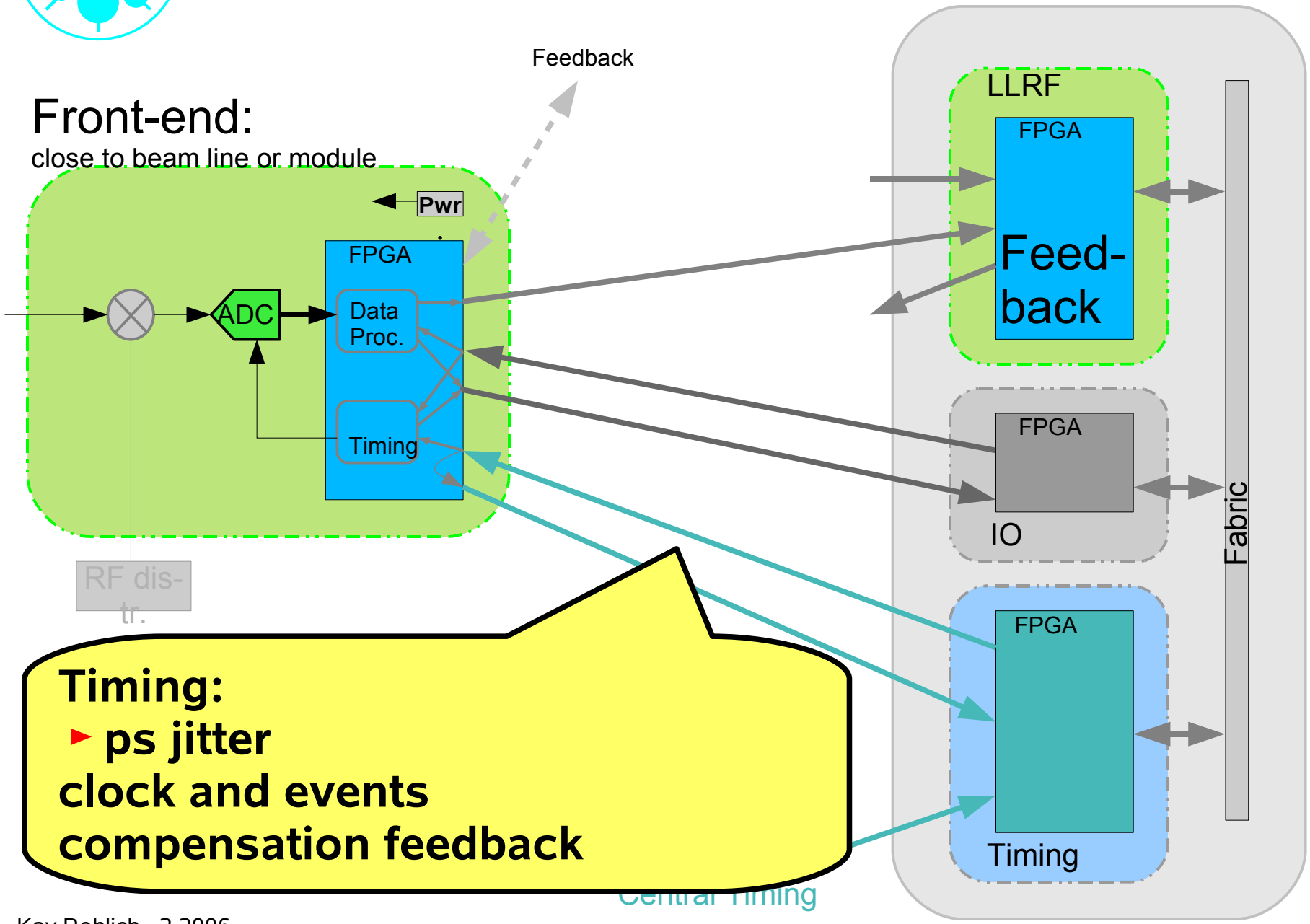
Gbit Links: Controls Data



Control Flow:

- **different data types:**
  Programs incl. FPGA, parameters, measurements ..
- **bidirectional**

Gbit Links: Timing

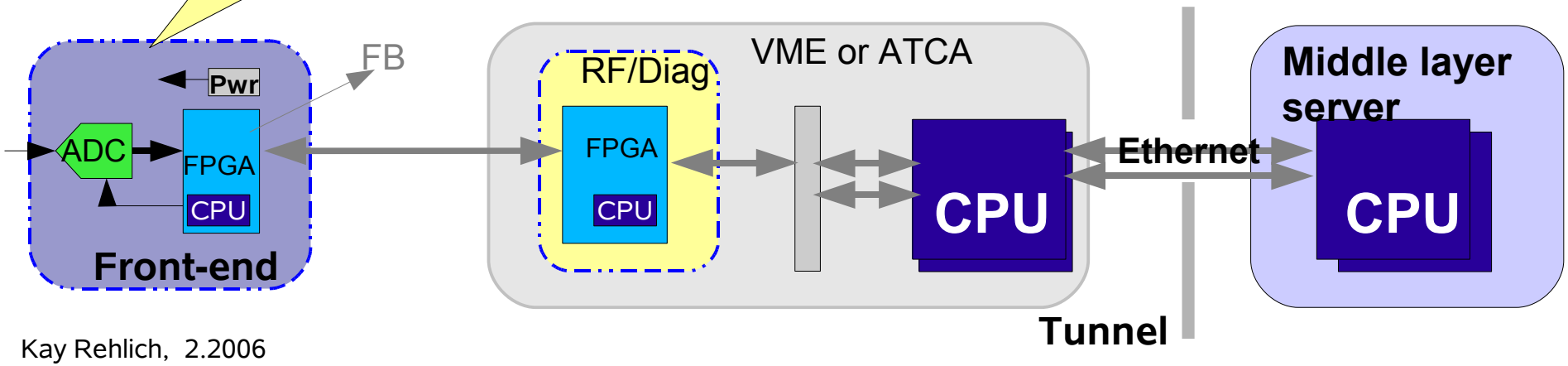


Timing:
- **ps jitter**

clock and events
compensation feedback

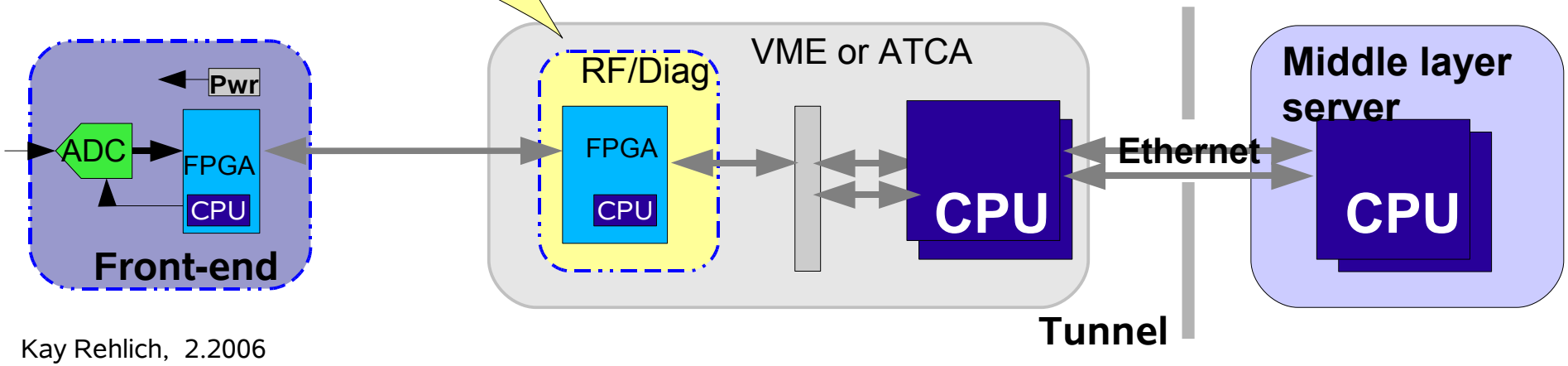
Task Distribution (1)

Preprocessing: **reduce data** to e.g. one set per bunch
channel based calculations (e.g. rotation matrix, sub vector sum)
Send data to 'collector' [and additional fast feedback systems]
Timing: adjust clock and triggers
Program loading and verification
Radiation recovery / tolerance
Self-tests / calibrations
Provide debug data (e.g. full samples) on request



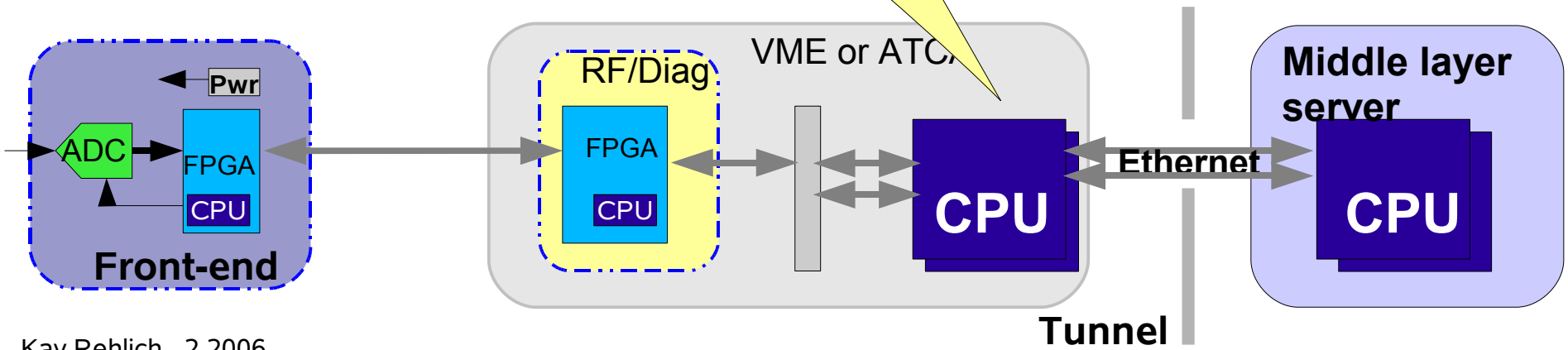
Task Distribution (2)

Feedback algorithm, feedforward, state estimator
Diagnostics data processing (..loaded Q..)
Data buffer for control system
Exception handling and quench detection
Table generation from simple parameters from control system
Program loading and verification
Self-tests



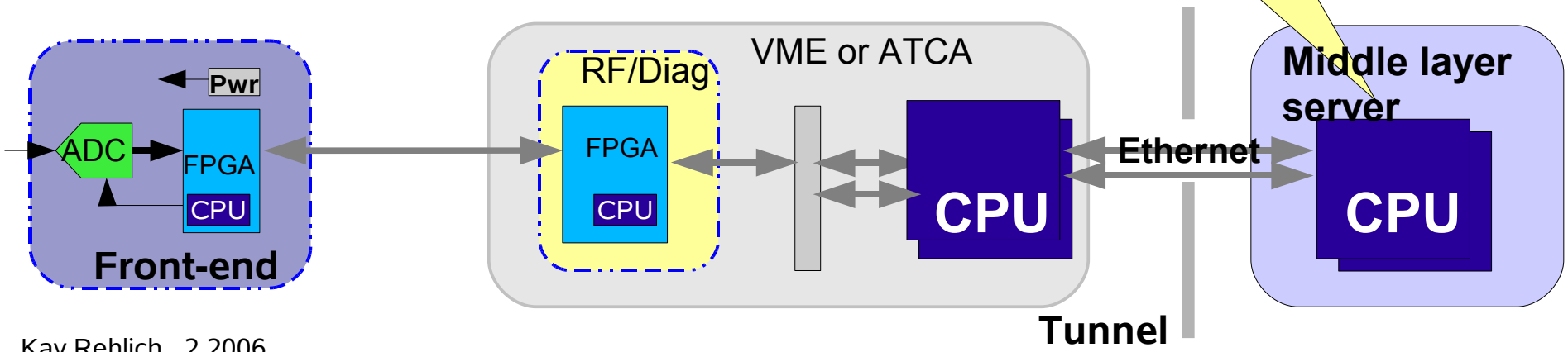
Task Distribution (3)

Data buffer for the control system
'self-healing' of the feedback and diagnostics controllers:
check/restart programs, parameters and processors
State machines
'fail-over' control (e.g. feedforward only operation)
Redundant CPU possible (ATCA)
Auto tuning cavity f, adaptive feedforward, loop phase, calibrations



Task Distribution (4)

Global services (for multiple devices)
Alarming
Automation (start-up, optimization, calibration, operation..)
Data Acquisition
Slow feedbacks
Global momentum manager or global orbit



Task Distribution, conclusions

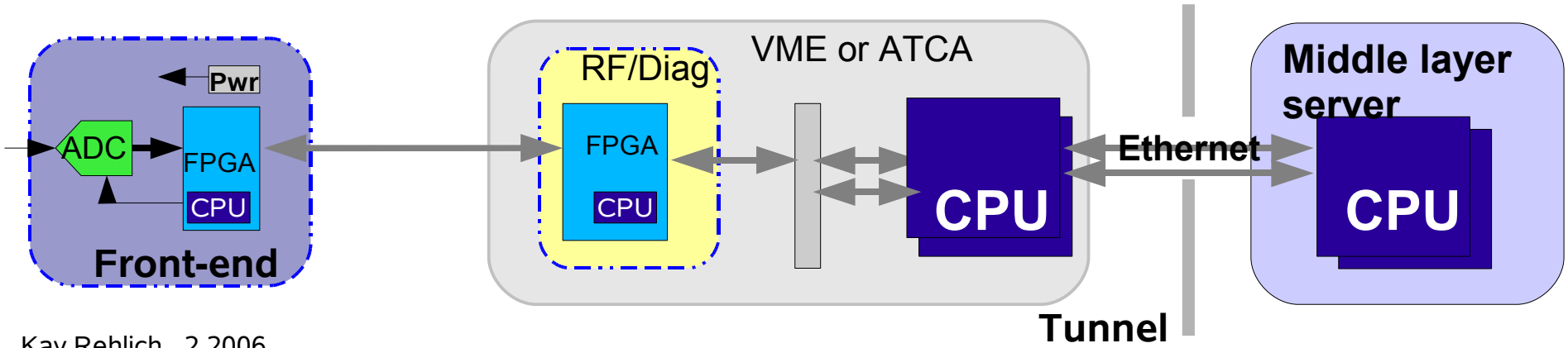
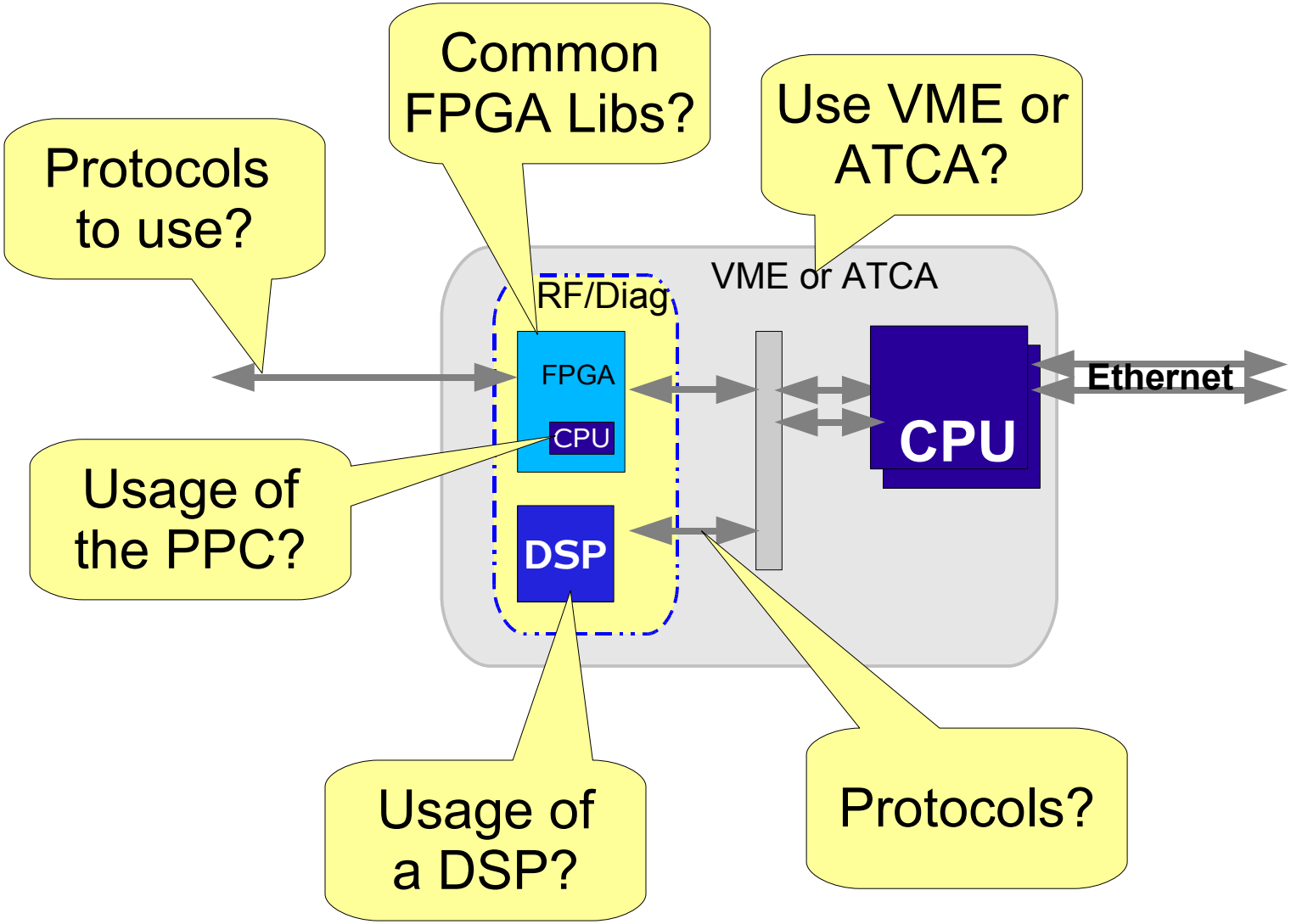
Where to implement programs ?

Complexity, maintainability of programs

redundancy (self-healing)

High data throughput

Less dependency of other devices



Some Open Questions