ATLAS
DAQ ↔ DCS
Communication
(DDC)
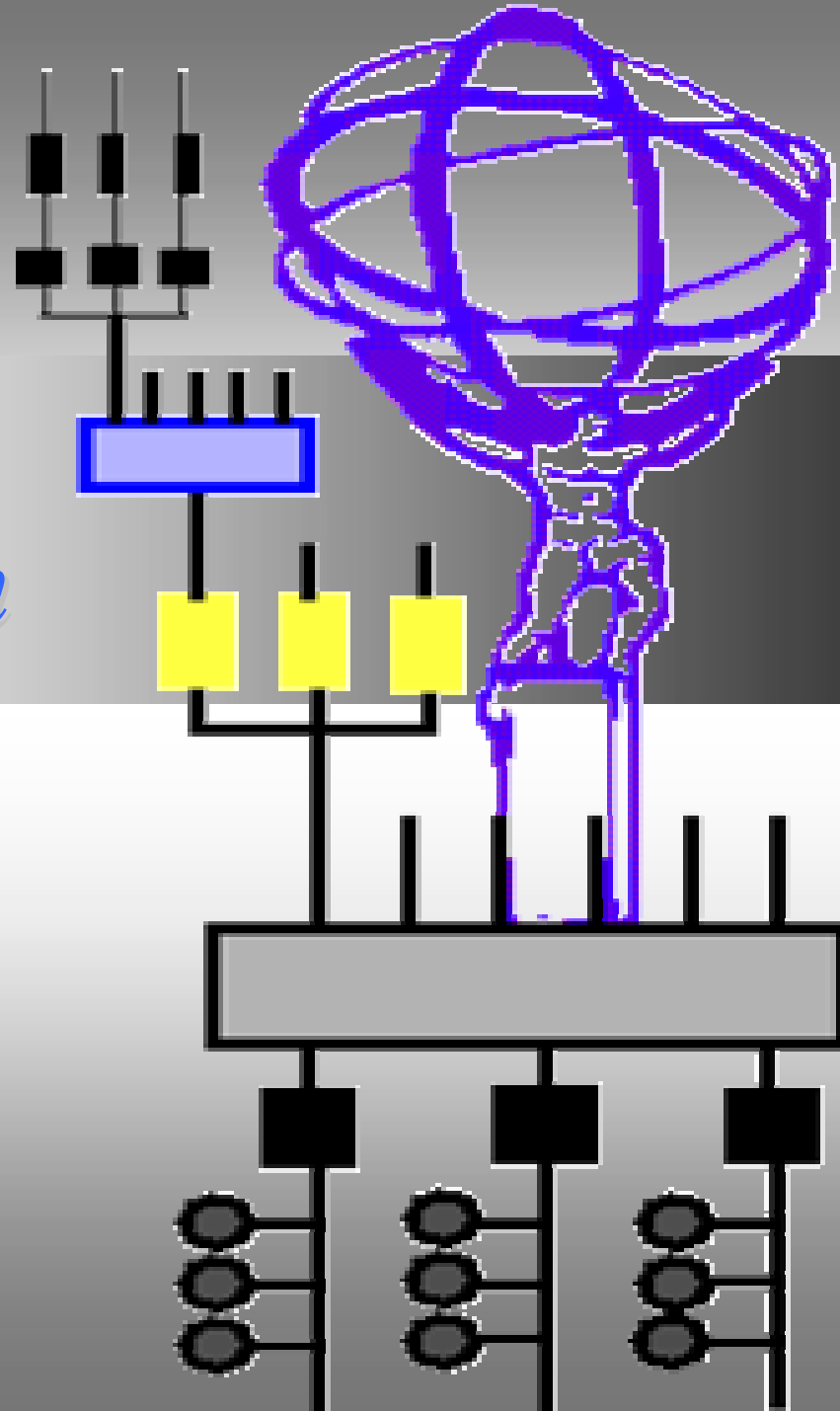
Robert Hart
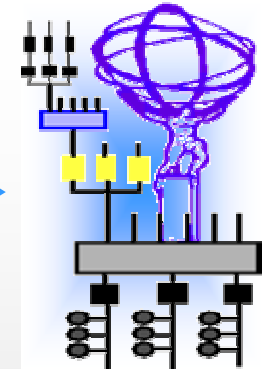
NIKHEF, Amsterdam

CHEP'01 Beijing

Why DDC ?
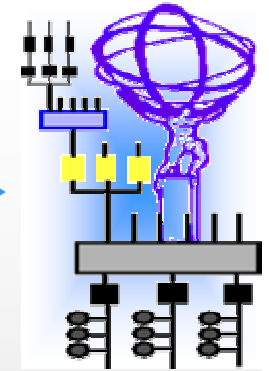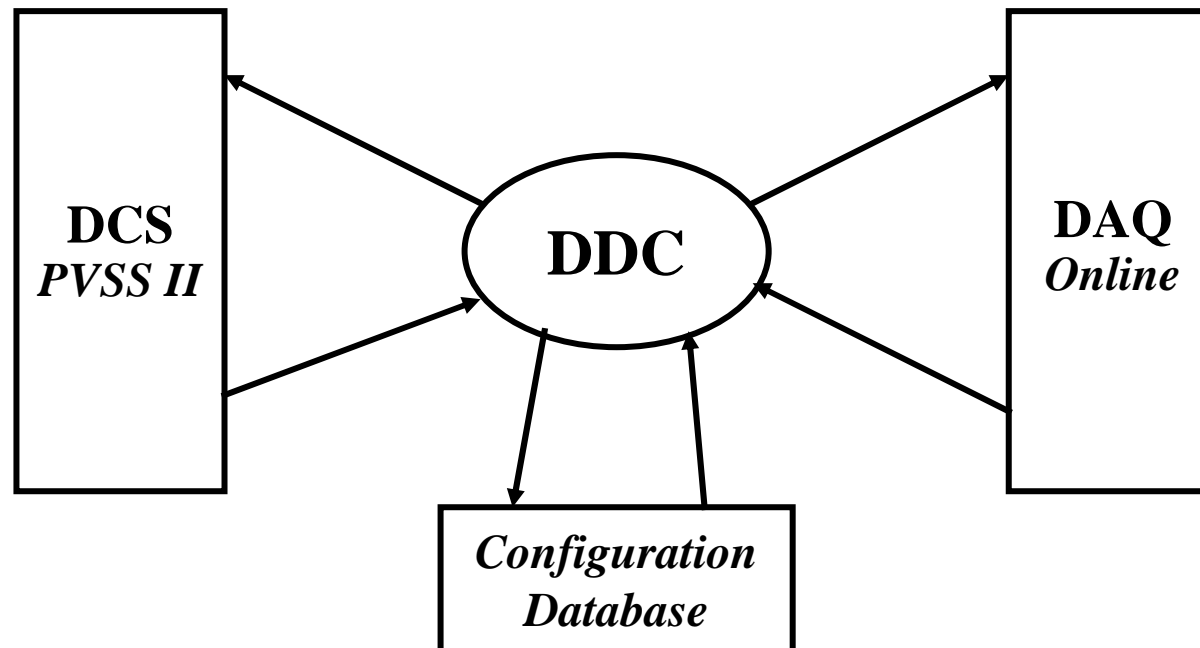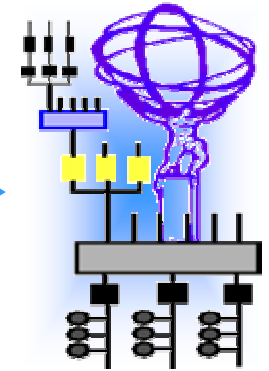
- **ATLAS: control paths DAQ & DCS are separated (physical and logical)**

- **Communication is needed:**

  - **Data exchange (parameters, status)**

  - **Messages like alarms from DCS $\rightarrow$ DAQ**

  - **Commands from DAQ $\rightarrow$ DCS**

- **Goal of DDC:**

  Interface between DAQ & DCS
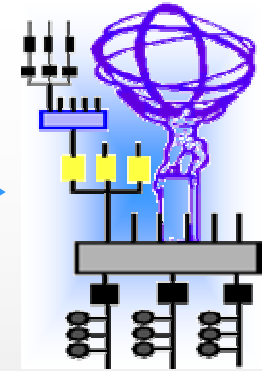
DDC domain decomposition

- **DDC-DT: Bi-directional exchange of data like parameters and status values.**

- **DDC-MT: Transmission of DCS messages (*alarms*) to DAQ.**

- **DDC-CT: Ability for DAQ to issue commands on DCS (*load, start, run*, etc).**

- **Interface points:**
  - **DCS: PVSS II**
  - **DAQ: Online**

DDC Context Diagram


```
┌──────────┐                              ┌──────────┐
│   DCS    │ ←─────        ────→           │   DAQ    │
│ PVSS II  │       ╲      ╱                │  Online  │
│          │    ┌────────────┐            │          │
│          │ ──→│    DDC     │←──          │          │
│          │    └────────────┘            │          │
└──────────┘      ↑↓   ↑                  └──────────┘
                ┌──────────────┐
                │Configuration │
                │  Database    │
                └──────────────┘
```
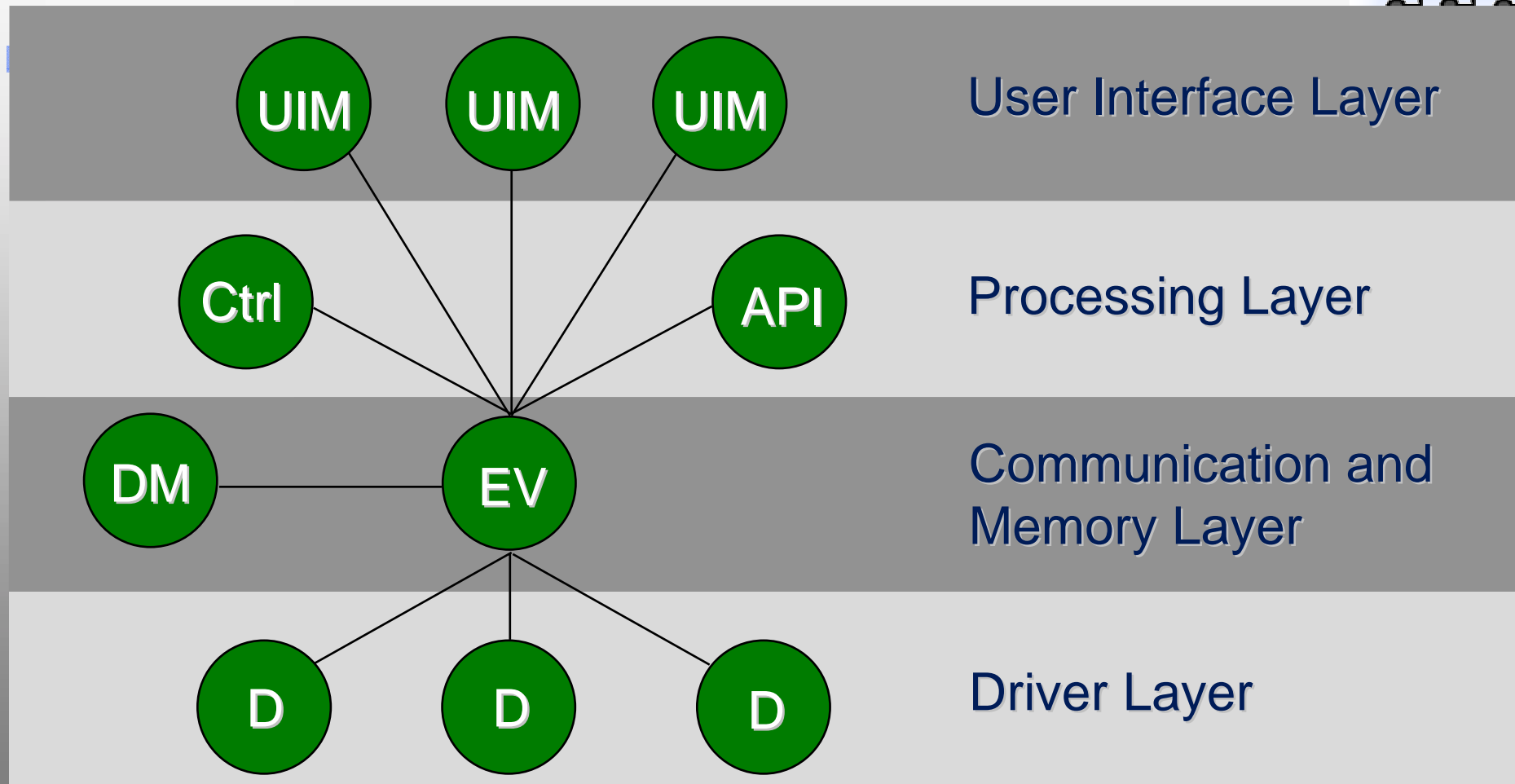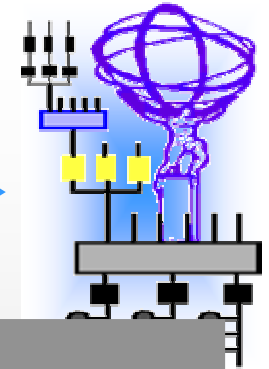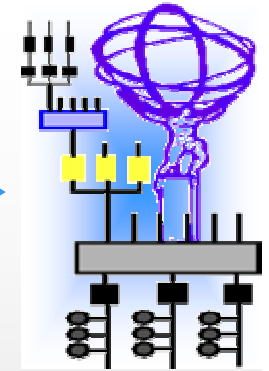
Design Prerequisites

- **Any manipulation with the physics data is beyond the scope of the DDC software.**

- **The DCS is expected to be operational at all times.**

- **<u>Partitioning</u>: DDC is aware, but not responsible !** The concept is known to both DAQ and DCS and should be compatible in terms of boundaries and locking of resources.

  - **DAQ: a subset of the experiment capable to run independently.**

  - **DCS: vertical slice, which controls a subsystem, defined as an arbitrary part of the detector.**

PVSS II Architecture

UIM UIM UIM **User Interface Layer**

Ctrl API **Processing Layer**

DM EV **Communication and Memory Layer**

D D D **Driver Layer**

PVSS II Architecture

- **API Manager**
  - **C++ class library**
  - **Interface to external world, like DDC**

- **Event Manager (EV)**
  - **Administers *datapoints* and events**
  - **Datapoint: basic data container (simplex and complex)**

- **Control Manager**
  - **Complex processing of data points**
  - **Interpreter with C++ syntax (NB: indexing starts at 1 !!!)**

- **Platforms: WNT & Linux !**
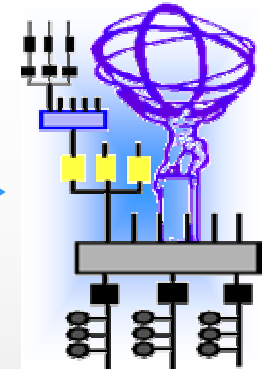
Online Software
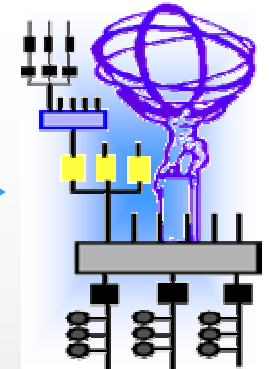
- ● **Subsystem of ATLAS DAQ/EF Prototype-1**
  - ● **Others: Detector Interface, Data Flow, Event Filter**

- ● **Software for Configuring, Controlling and Monitoring the DAQ system**
  - ● **Excludes the processing and transportation of physics data**
  - ● **General Purpose "Control System": *Open Source***
  - ● **The Software *glue* of the Experiment**
  - ● **Primary programming language: C++**
  - ● **Internal communication: CORBA (ILU)**

- ● **Platforms: Linux, LynxOS, Solaris**
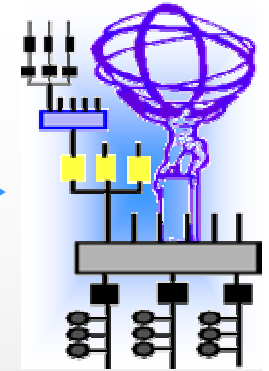
Online Components
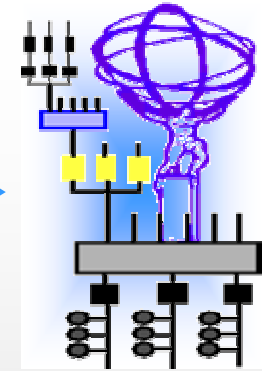
- **Run Control** — Controls configuration and data taking operations
- **Configuration Databases** — Defines all aspects of the configuration
- **Information Service** — General purpose information exchange
- **Message Reporting System** — Report/capture of error/information messages
- **Process Manager** — Distributed job control of programs
- **Resource Manager** — Allows concurrent data taking activities
- **Integrated User Interface** — Gives current status and control to shift operator
- **Online Bookkeeper** — Electronic tape log book
- **Test Manager** — Bank of functionality tests for hardware and software
- **Diagnostics System** — Uses tests in the Test Manager to diagnose problems
- **Event Dump** — Event monitoring program with GUI
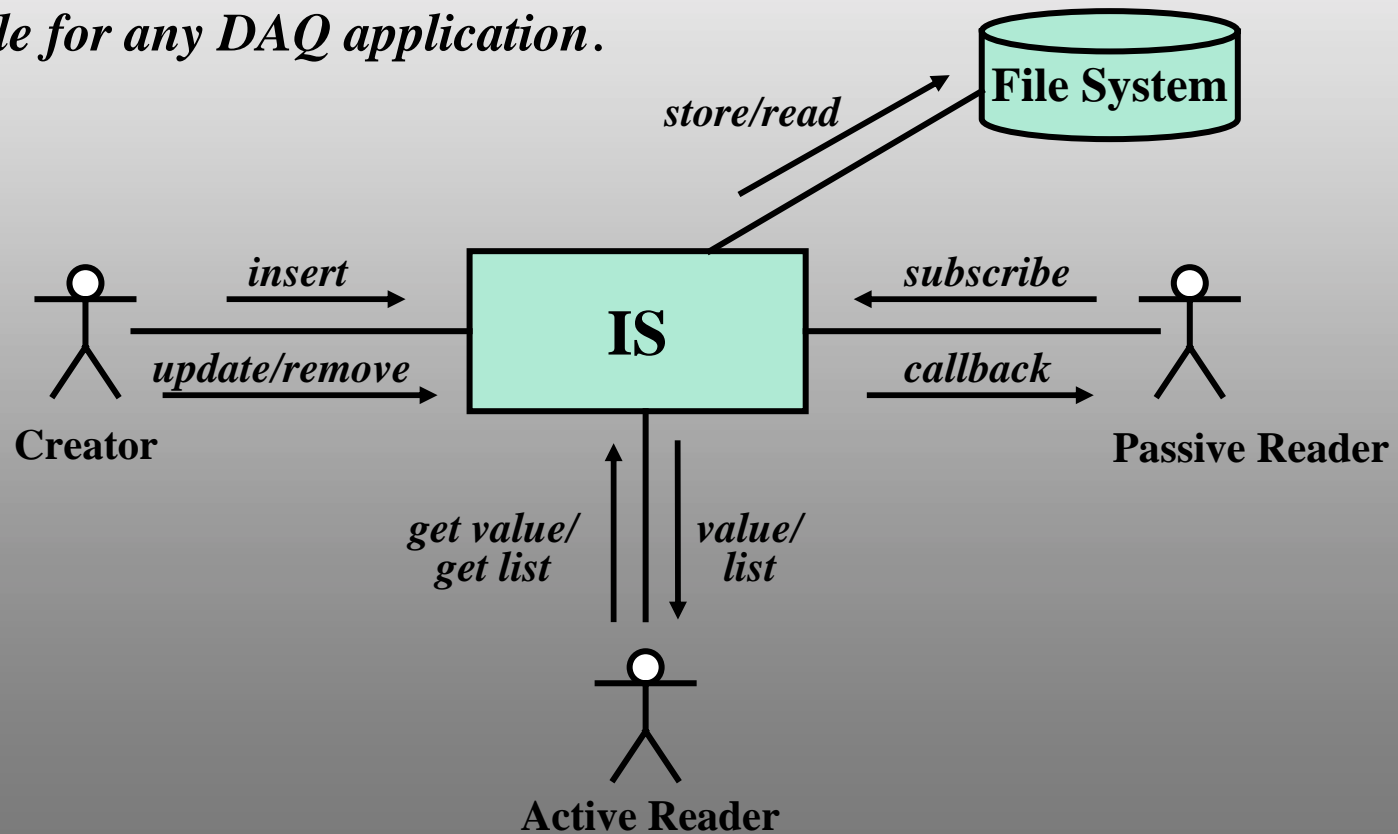
DDC & Online

- **DDC_DT: Data Exchange Facility** (DAQ ↔ DCS)
  - **Information Service (IS) component**
- **DDC_MT: Message Passing Facility** (DCS → DAQ)
  - **Message Reporting System (MRS) component**
- **DDC_CT: Passing Commands** (DAQ → DCS)
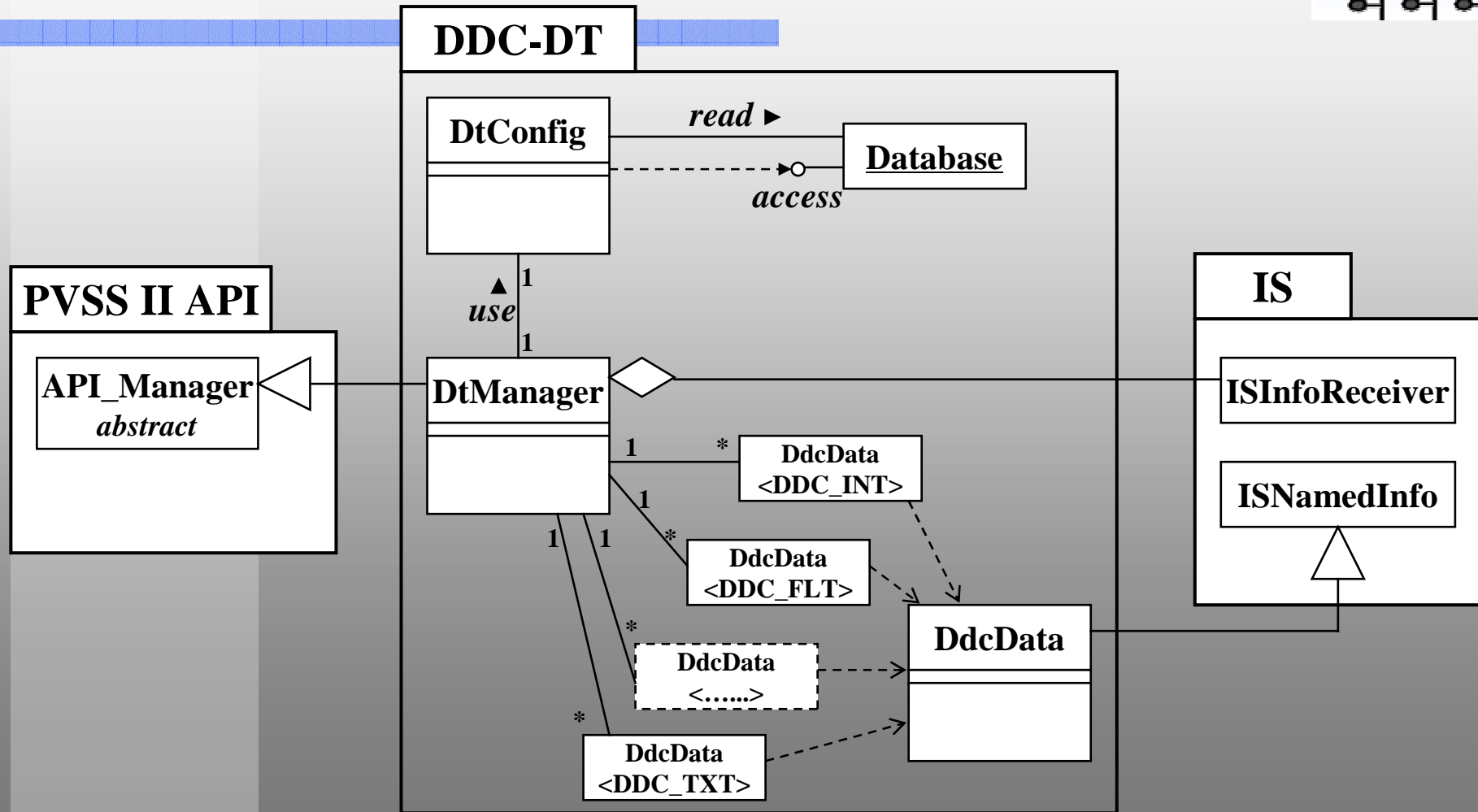 - **Run Control (RC) component**

Information Service (IS)

Goal: *A persistent information storage system available for any DAQ application.*

File System

store/read

insert

update/remove

IS

subscribe

callback

Creator

Passive Reader

get value/ get list

value/ list

Active Reader

DDC-DT Class Diagram

DDC-DT

DtConfig
read ▶
Database
access

PVSS II API

use 1 / 1

API_Manager
abstract

DtManager

IS

ISInfoReceiver

ISNamedInfo

1 / * **DdcData <DDC_INT>**

1 / * **DdcData <DDC_FLT>**

DdcData <......>

DdcData <DDC_TXT>

DdcData

DT: Collaboration Diagram I

```
        ┌─┬─┐                              ┌─┬─┐
        └─┴─┘                              └─┴─┘
    ┌──────────┐                     ┌──────────┐
    │  SCADA   │                     │   DAQ    │
    │ Database │                     │Application│
    └──────────┘                     └──────────┘
         ▲  │                             │  ▲
       1 │  │                           4 │  │ 5
  subscribe│  │data value      subscribe│  │data value
         │  │ 2                          │  │
    ┌─┬─┐│  │                       ┌─┬─┐│  │
    └─┴─┘│  ▼                       └─┴─┘│  ▼
    ┌──────────┐   data value  3   ┌──────────┐
    │  DDC-DT  │ ─────────────────>│    IS    │
    └──────────┘                   └──────────┘
```
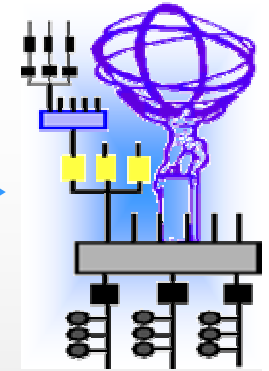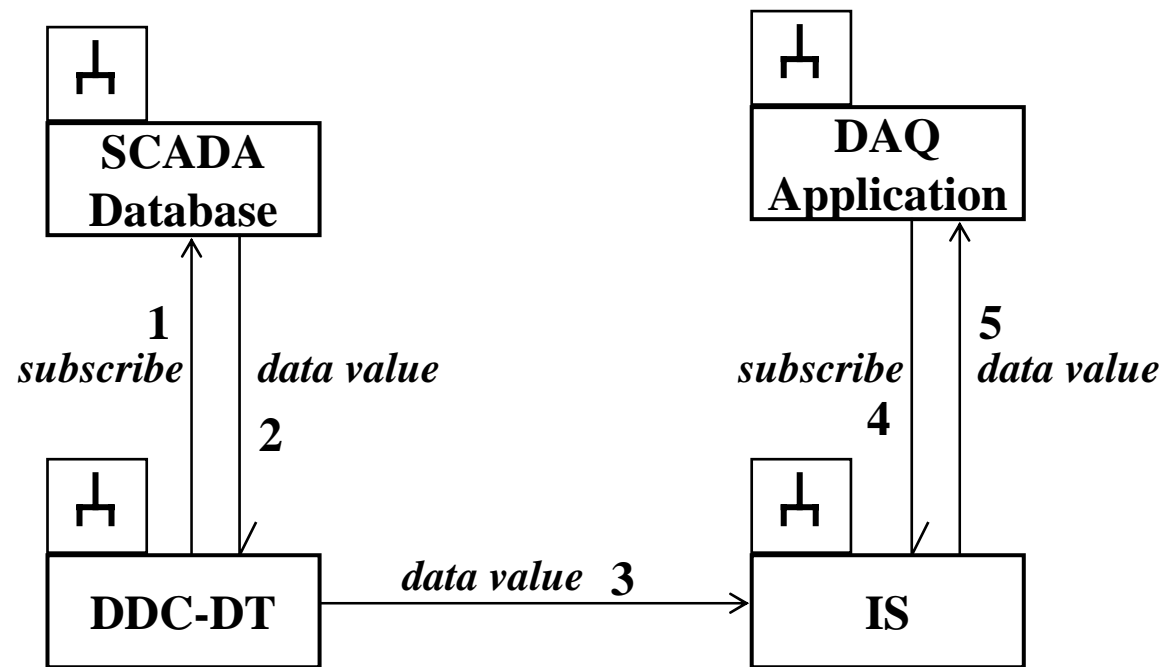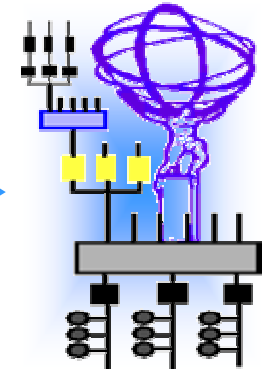
DCS →DAQ data transfer

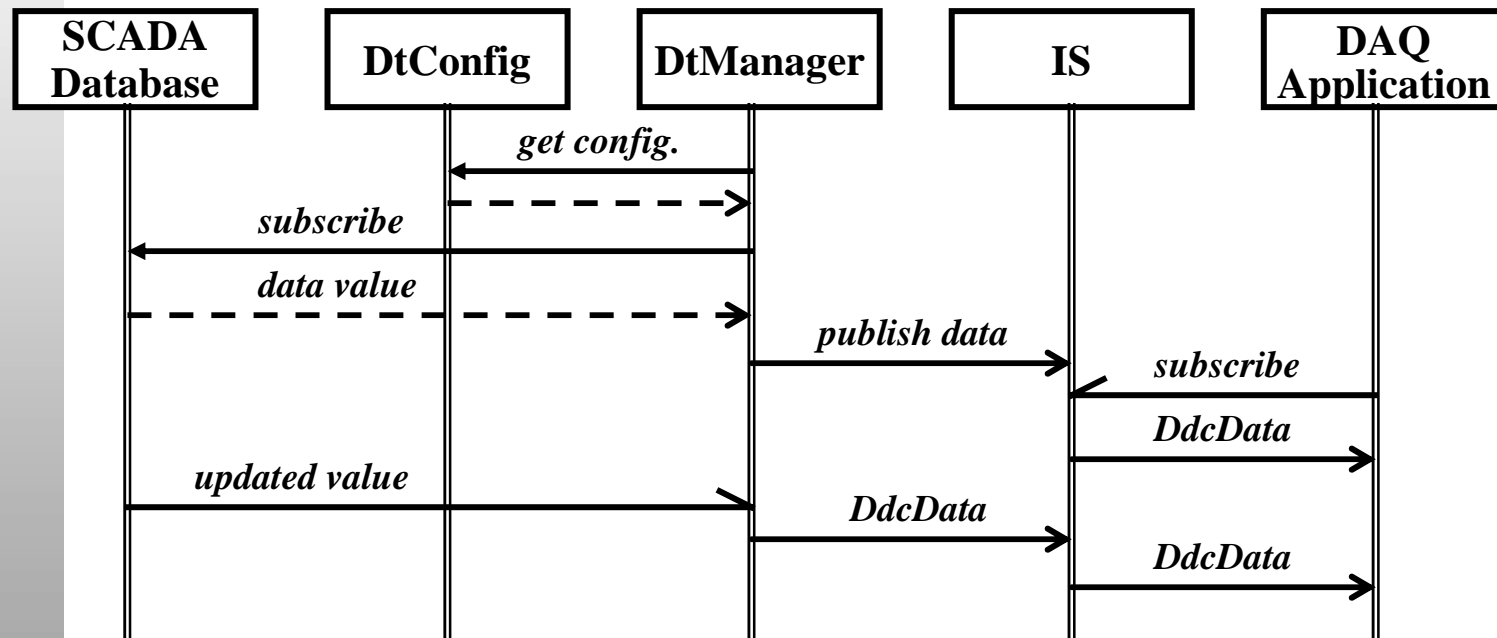# DT: Collaboration Diagram II

```
            SCADA                        DAQ
           Database                   Application

         1      5
   set "unknown"  data value        publish data
                                              3

                  subscribe   2
        DDC-DT  ←─────────────────→  IS
                    4   data value
```
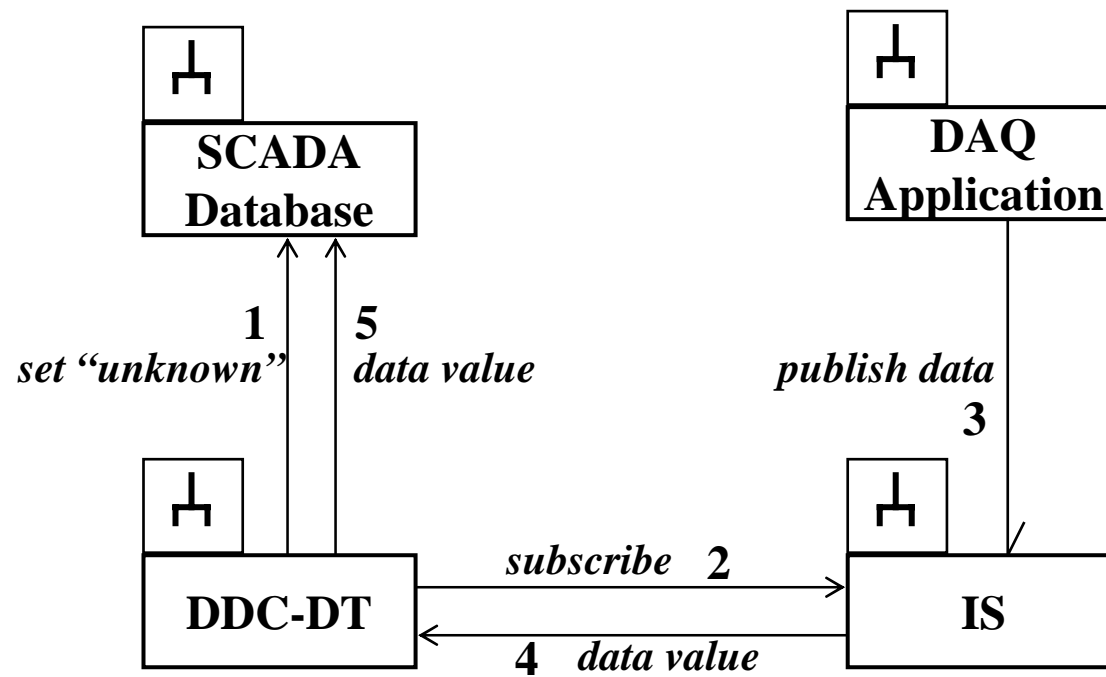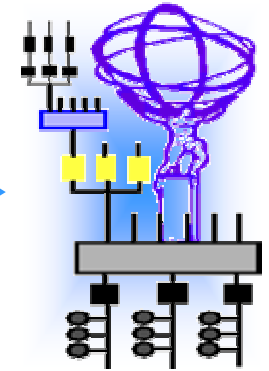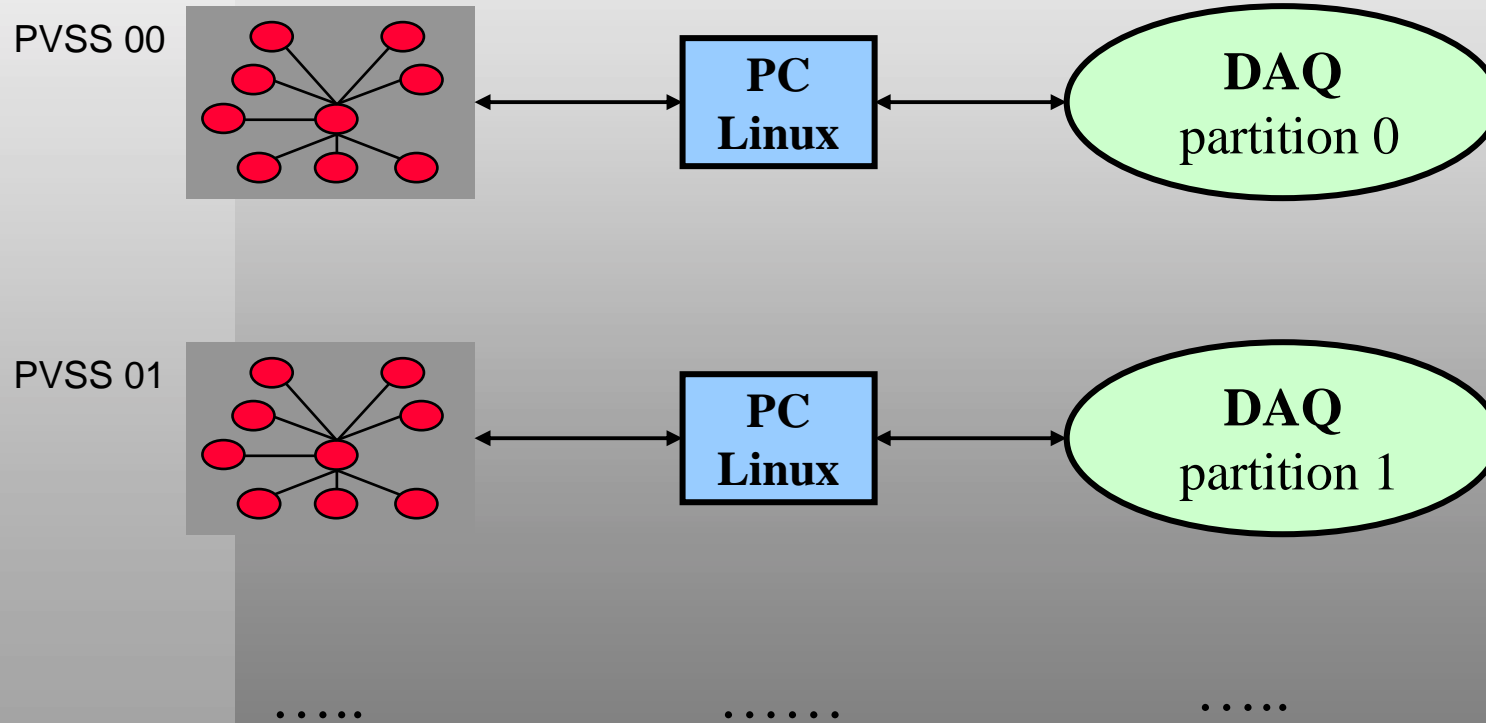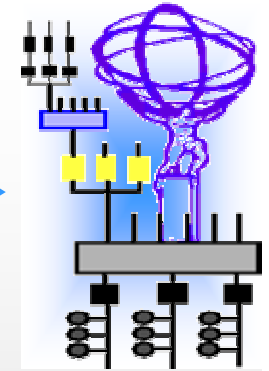
DAQ →DCS data transfer

DT: Sequence Diagram
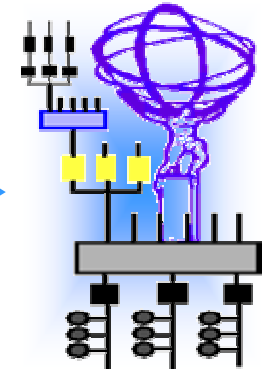
DCS → DAQ data transfer

Current Implementation

PVSS 00

```
[red node cluster]  <——>  | PC
                          | Linux |  <——>  ( DAQ
                                            partition 0 )
```

PVSS 01

```
[red node cluster]  <——>  | PC
                          | Linux |  <——>  ( DAQ
                                            partition 1 )
```

…..   ……   …..

Online Software Process

Problem Statement

↓

Analysis

↓

Design

↓

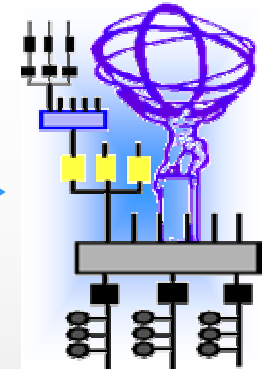**Implementation**

↓

Testing

↓

Maintenance
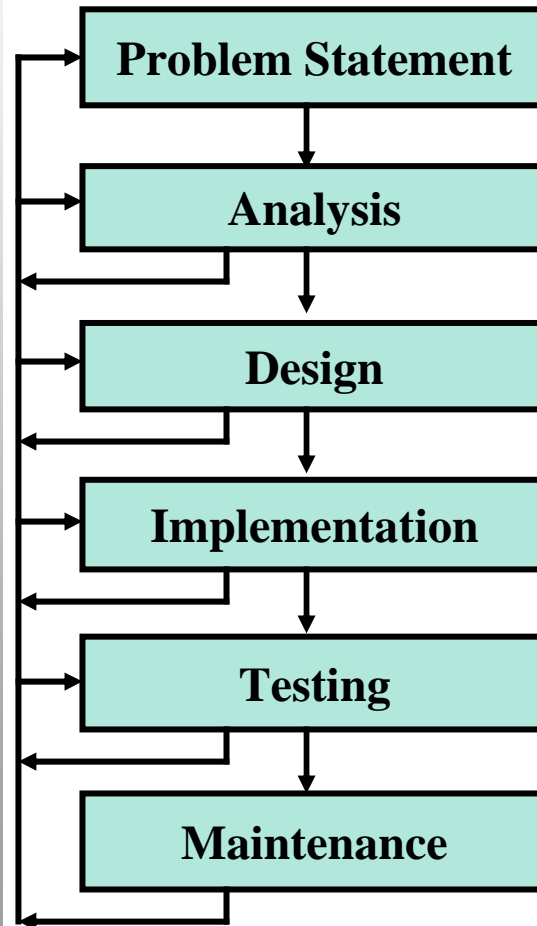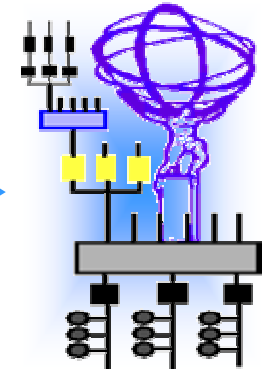
- **Software Lifecycle**
- **Documents:**
  - **URD**
  - **HLD**
  - **Test Plan**
  - **Implementation Report**
  - **User Guide**
- **All documents and code <u>inspected.</u>**
- **Benefits of inspection:**
  - **Detecting faults asap.**
  - **Sharing knowledge.**
  - **Learning and knowing each other (both authors & inspectors!)**

DDC wrap-up

- **DDC: joint ATLAS DAQ/DCS project**
  - **collaborators: CERN, NIKHEF, PNPI**
- **DDC: ready to use (release 0.2; DDC_MT not yet)**
- **Part of Online distribution (binaries, libraries, code & documentation) as full component**
- **Used at Tile-Calorimeter test-beam at CERN**
- **Main source of documentation:**
  - **ATLAS Online homepage:**
    http://atddoc.cern.ch/Atlas/DaqSoft/Welcome.html
  - **ATLAS DCS homepage:**
    http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/dcshome.html