CHEP06, 13-17 February 2006, Mumbai

LHCb Conditions Database

Marco Clemencic

marco.clemencic@cern.ch

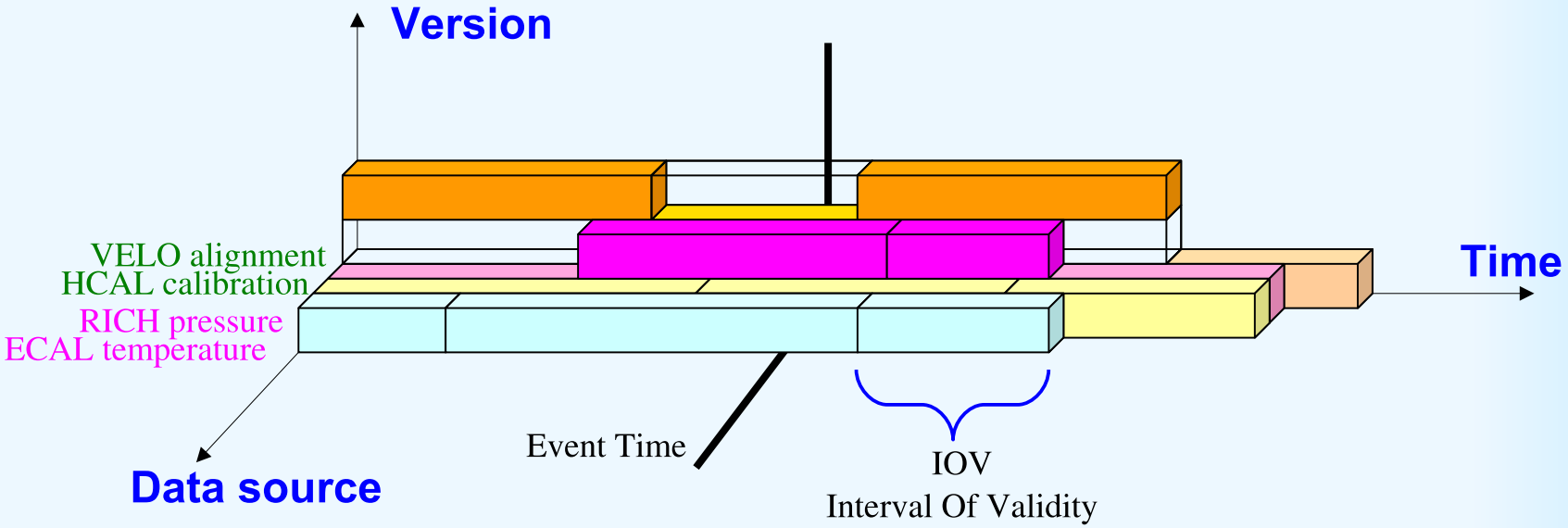
Overview

- Introduction
  - Definition of “Condition”
  - Requirements
- Conditions Database (CondDB)
- Update Mechanism
- Online Usage
- Deployment
- Summary

Introduction

Conditions

Time-varying non-event data



3 degrees of freedom:

- source
- time
- version

2 categories of conditions:

- off-line cond. (multi version)
- on-line cond. (single version)

LCG is developing a library to handle conditions: COOL (see A. Valassi's talk)

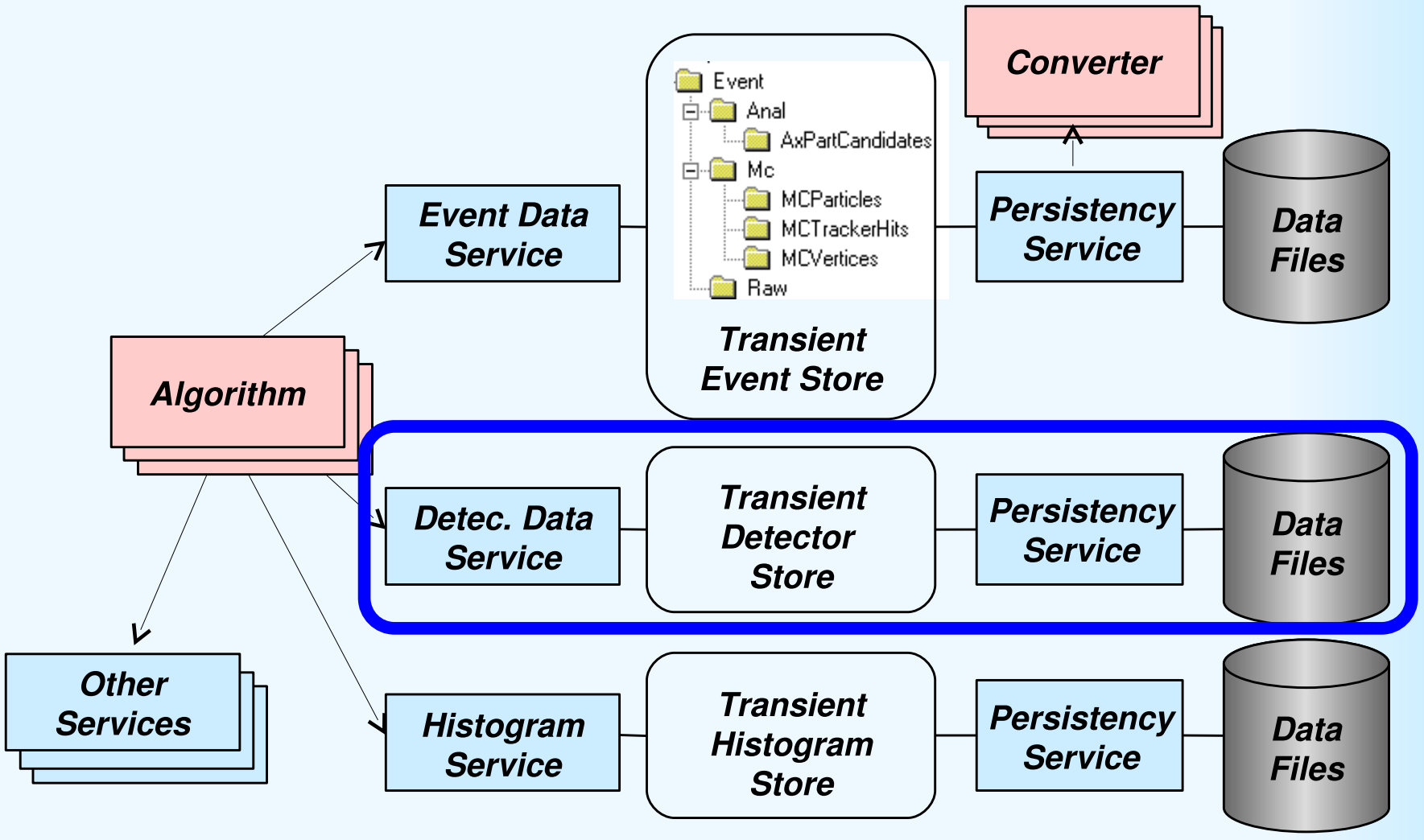
Requirements

An infrastructure is needed:

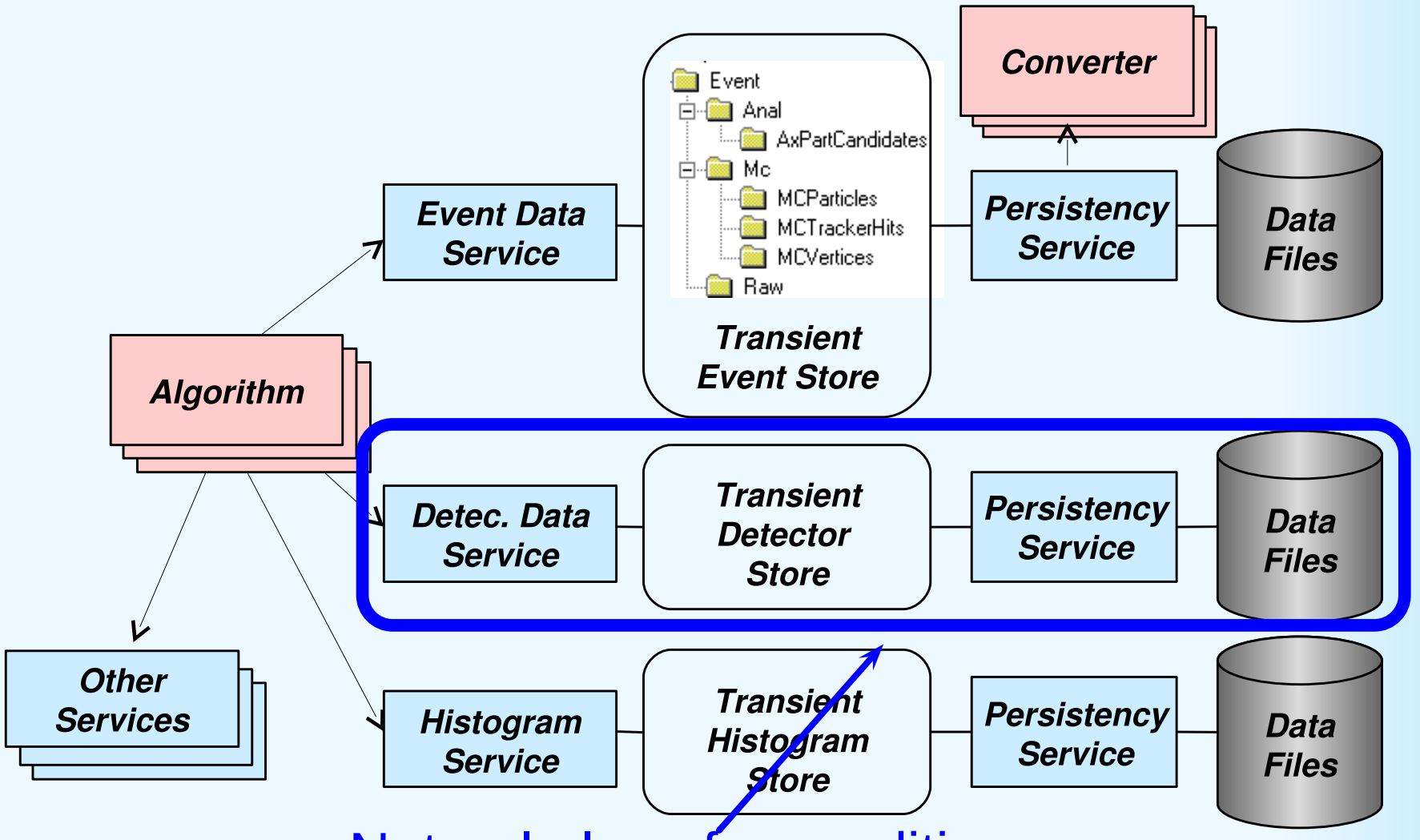
- Integrated in LHCb framework (Gaudi)
- Flexible $\rightarrow$ *freedom for the users*
- Efficient $\rightarrow$ *reduced overhead*
- Easy to use

Conditions Database

Gaudi Architecture



Gaudi Architecture

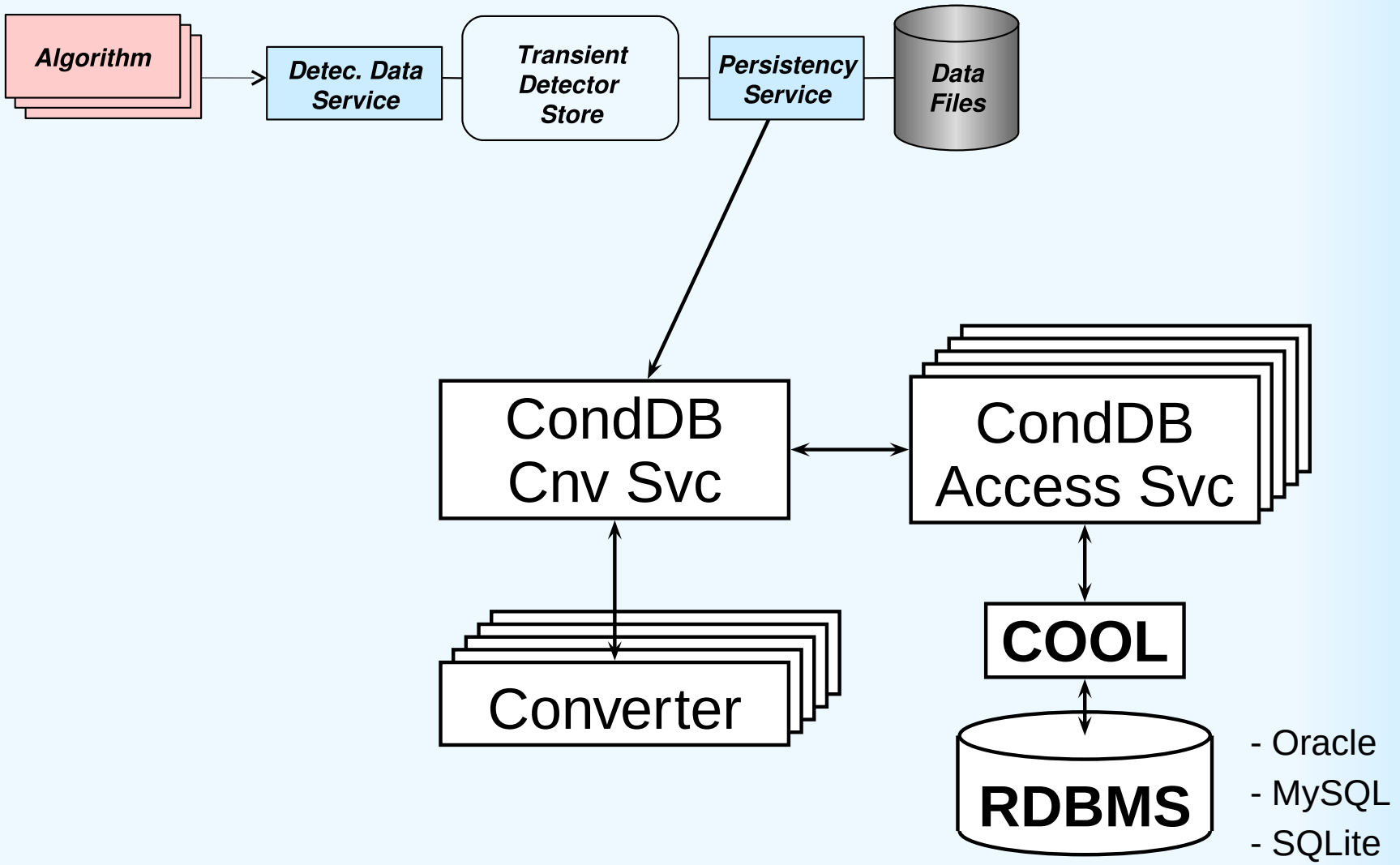


Natural place for conditions

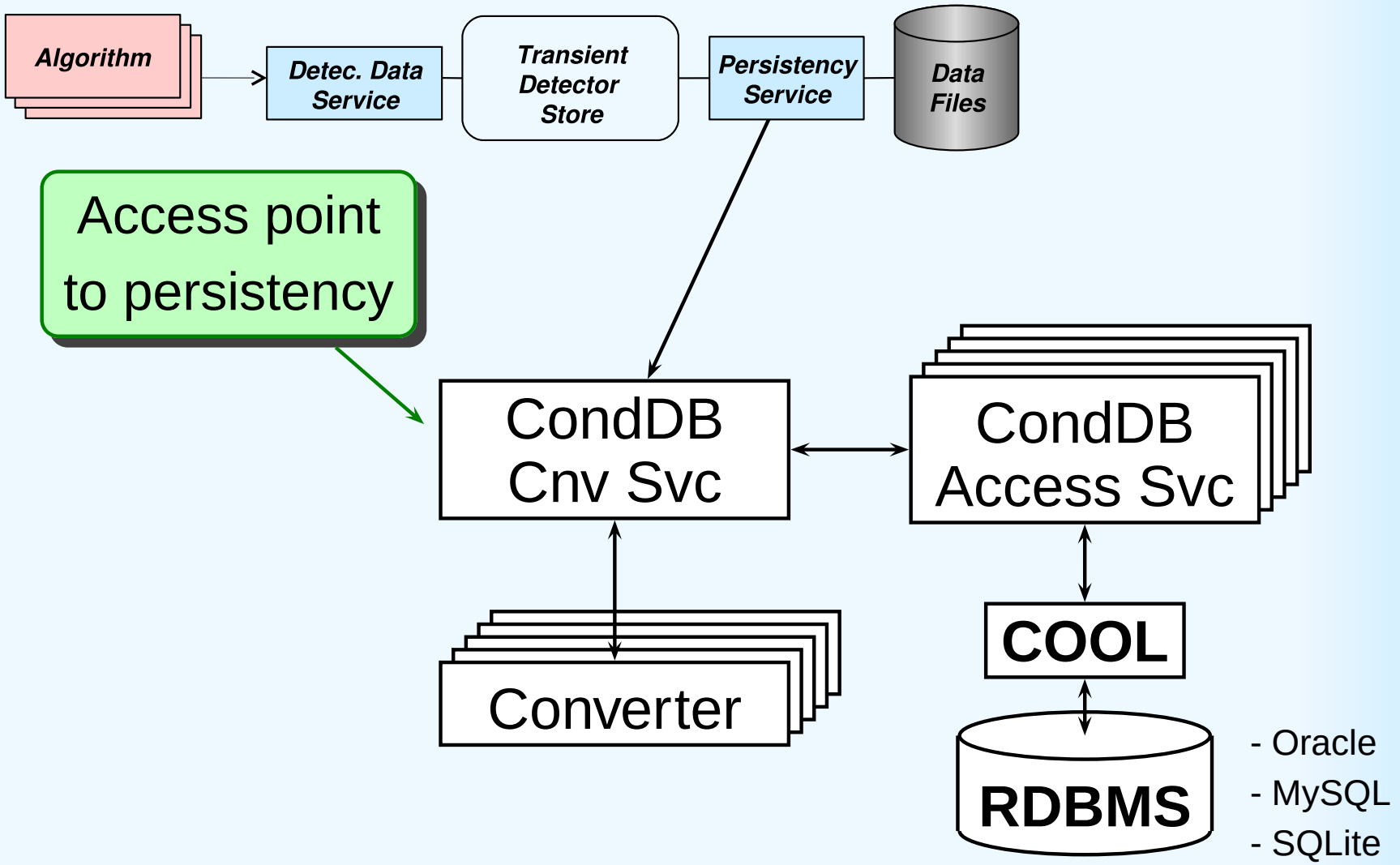
Detector Data Service

- Contains the “Detector Description”
  - classes providing detector informations (*main consumers of conditions*)
- Objects’ lifetime not depending on event loop
  - they can be *valid* for a set of events
- XML files for persistency
  - good compromise between human-readable and machine-readable

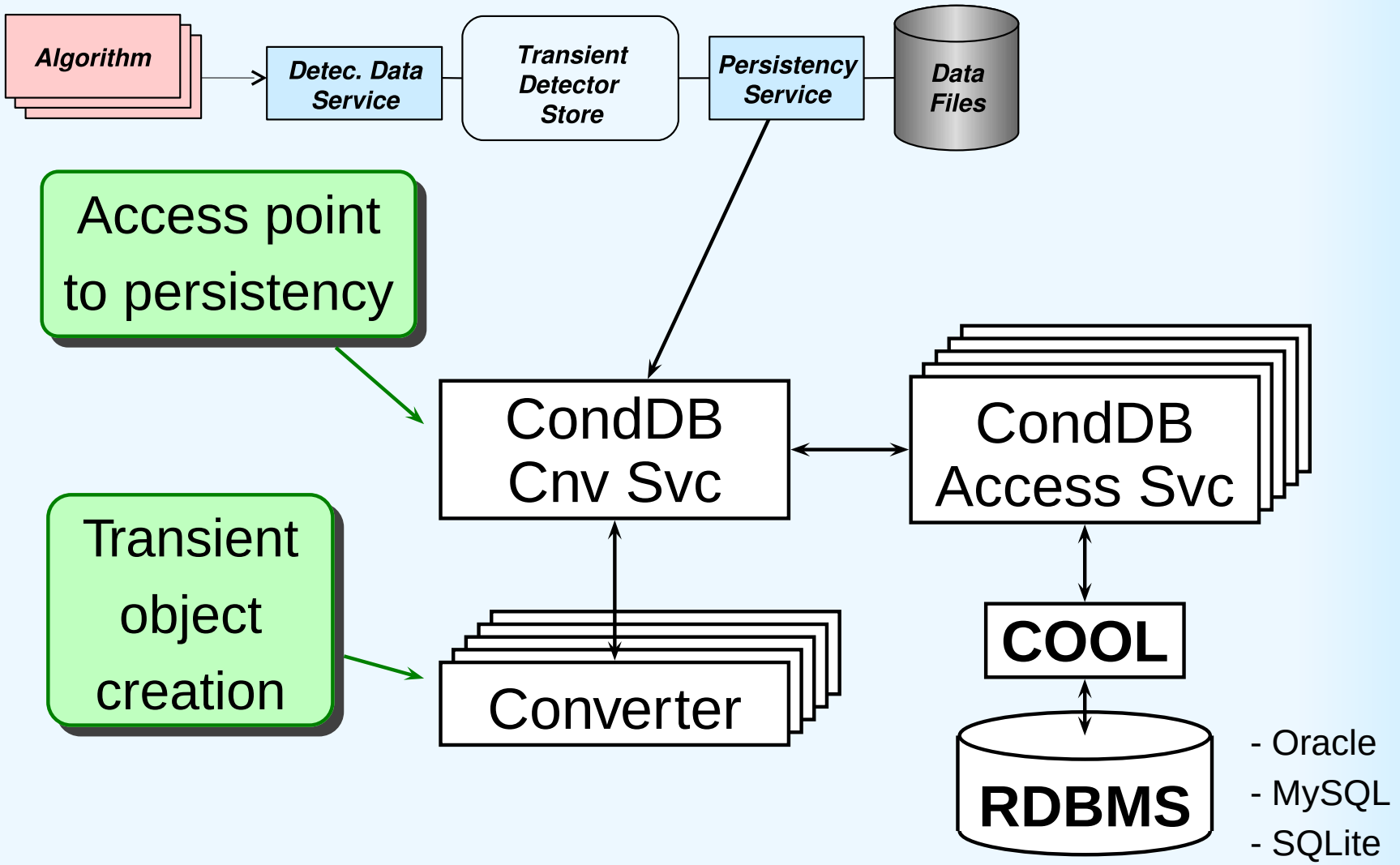
CondDB Conversion Service



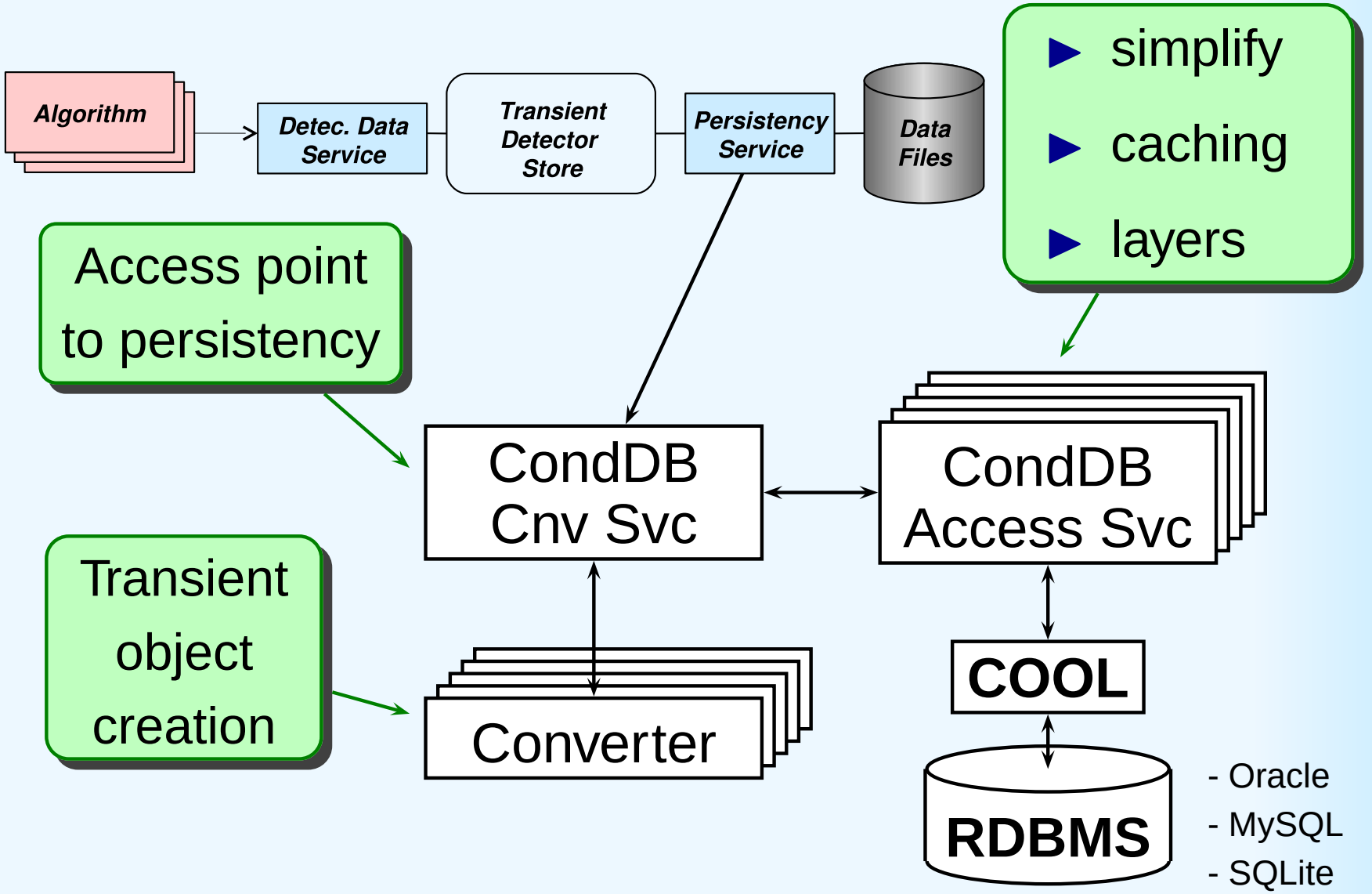
CondDB Conversion Service



CondDB Conversion Service



CondDB Conversion Service



First Tests

- Currently we do not have any real data to put into the database
- To estimate the performances we create a copy of all the detector description in a DB
- Loading all the data takes
  (on a 2.8 GHz Xeon)
  - $\sim$15 s from files
  - 1–2 min. from Oracle server
- There is still a lot of room for improvements

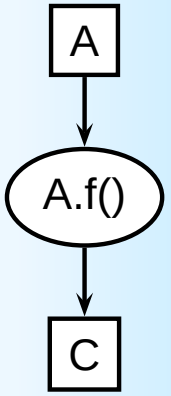
Update Mechanism

The Dependencies

- Object A depends on condition C

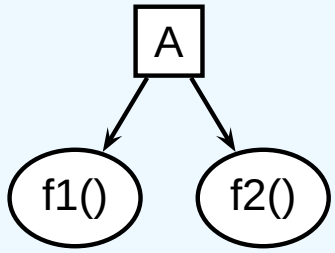
The Dependencies

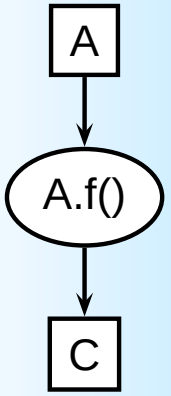
- Object A depends on condition C
  - when C changes, A does something
    (caching data, digest. . .)



The Dependencies

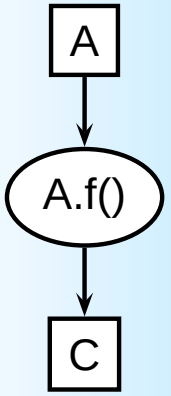
- Object A depends on condition C
  - when C changes, A does something
    (caching data, digest. . .)
- Flexibility
  - more actions

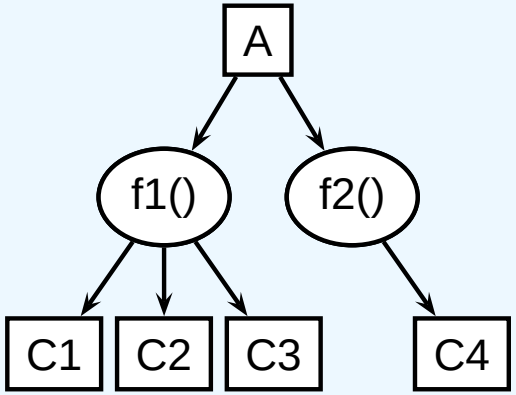




The Dependencies

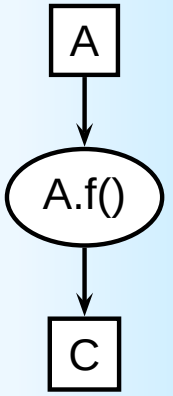
- Object A depends on condition C
  - when C changes, A does something
    (caching data, digest. . .)
- Flexibility
  - more actions
  - more conditions

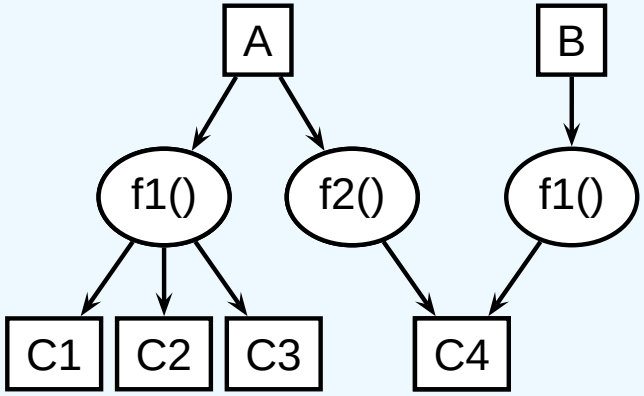




The Dependencies

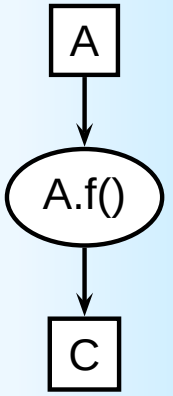
- Object A depends on condition C
  - when C changes, A does something
    (caching data, digest. . .)
- Flexibility
  - more actions
  - more conditions
  - more consumers

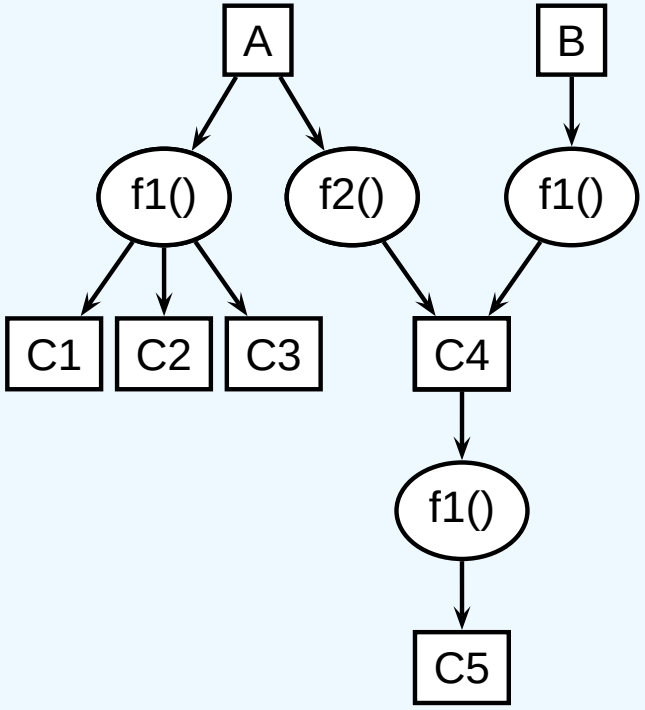




The Dependencies

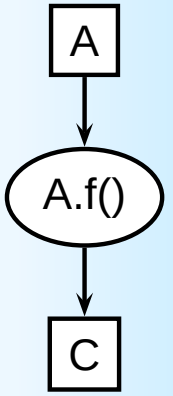
- Object A depends on condition C
  - when C changes, A does something
    (caching data, digest. . .)
- Flexibility
  - more actions
  - more conditions
  - more consumers
  - more levels

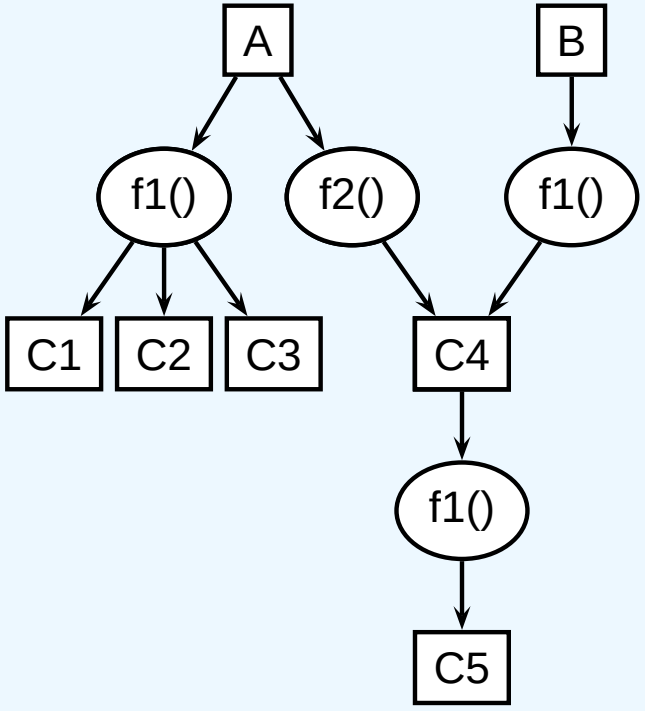




The Dependencies

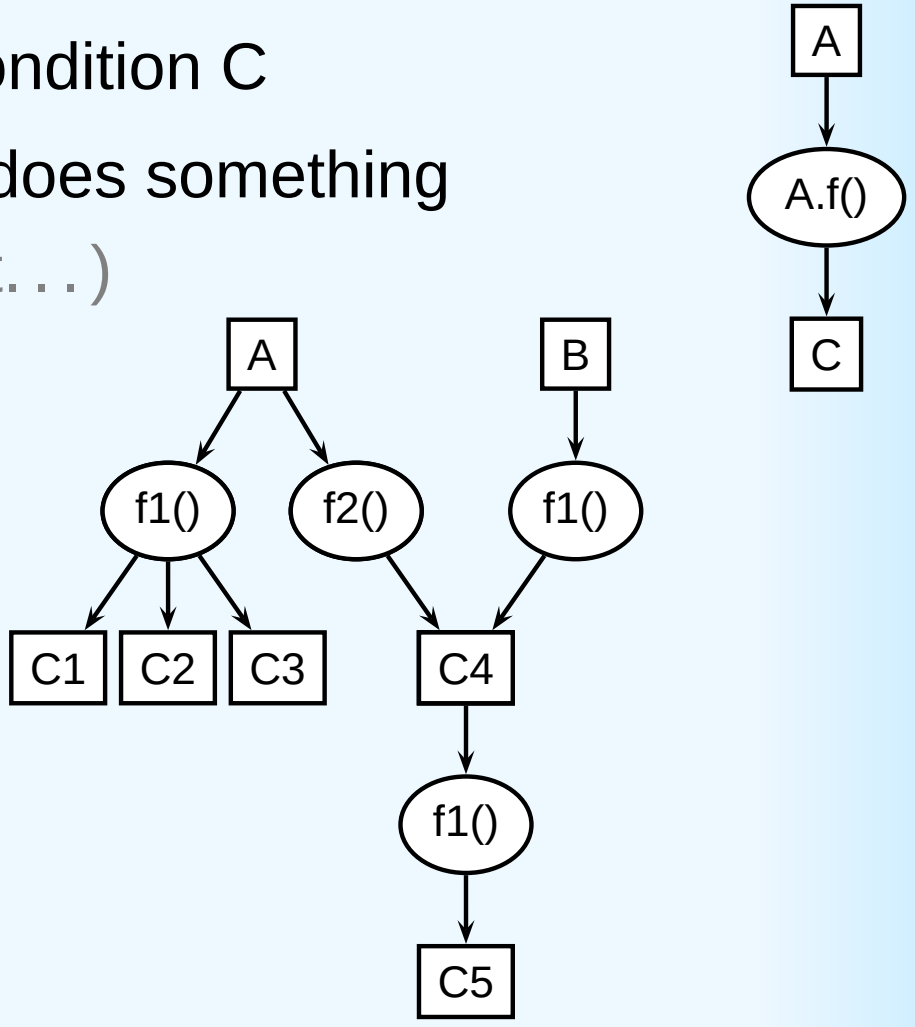
- Object A depends on condition C
  - when C changes, A does something
    (caching data, digest. . .)
- Flexibility
  - more actions
  - more conditions
  - more consumers
  - more levels
- Dynamic





The Dependencies

- Object A depends on condition C
  - when C changes, A does something
    (caching data, digest. . .)
- Flexibility
  - more actions
  - more conditions
  - more consumers
  - more levels
- Dynamic



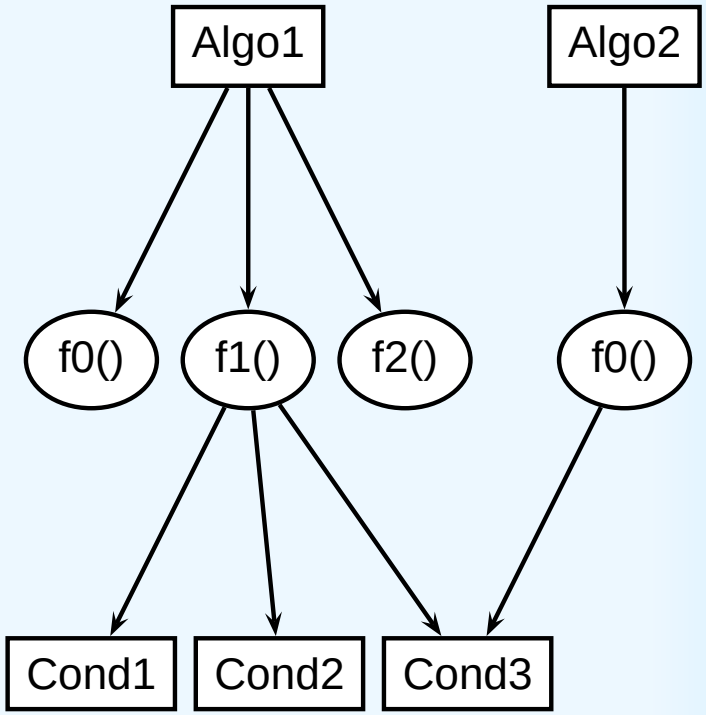
Network of Dependencies

Update Manager Service

- Specialized service dedicated to handle dependencies and updates
- At the begin of each event:
  - find objects needing an update
  - update objects
  - call user functions

Update Policy

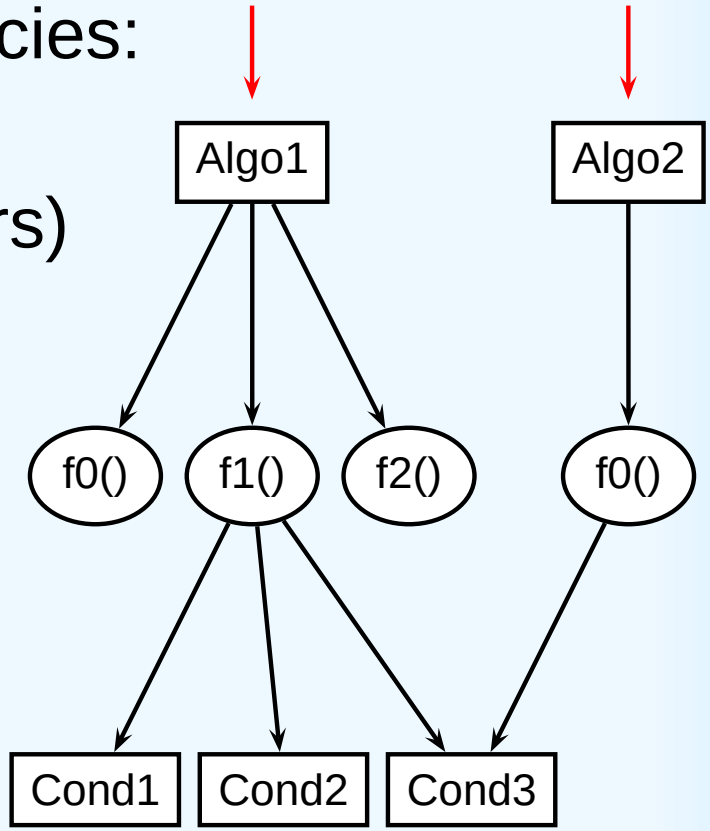
Given a network of dependencies:



Update Policy

Given a network of dependencies:

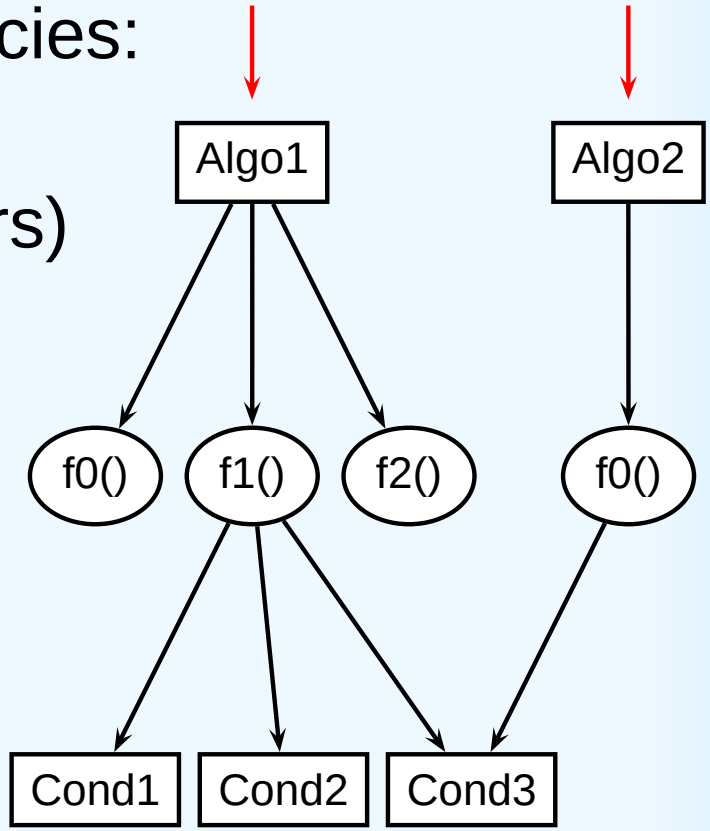
- start from the head
  (objects without consumers)



Update Policy

Given a network of dependencies:

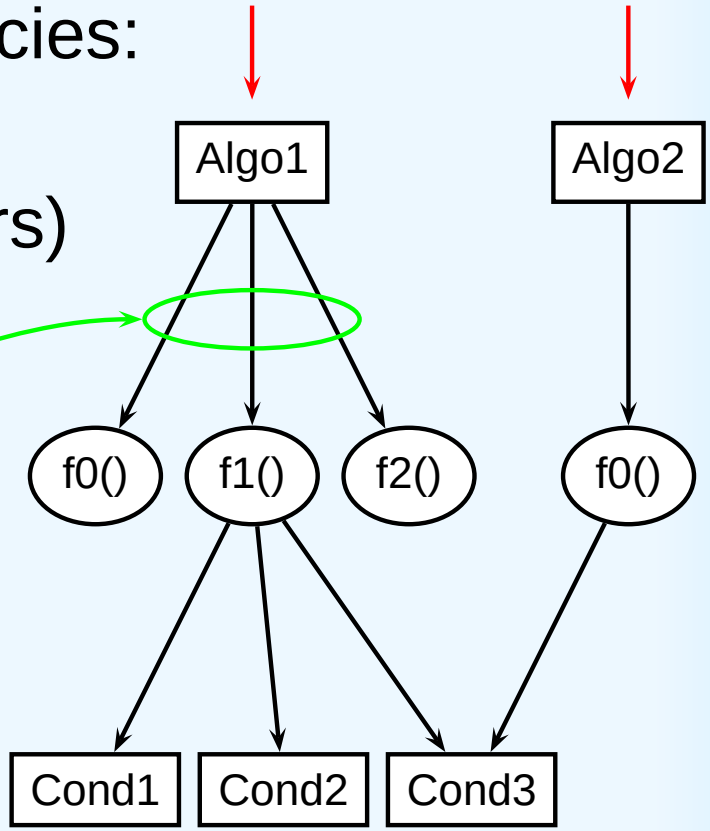
- start from the head (objects without consumers)
- intersection of validities



Update Policy

Given a network of dependencies:

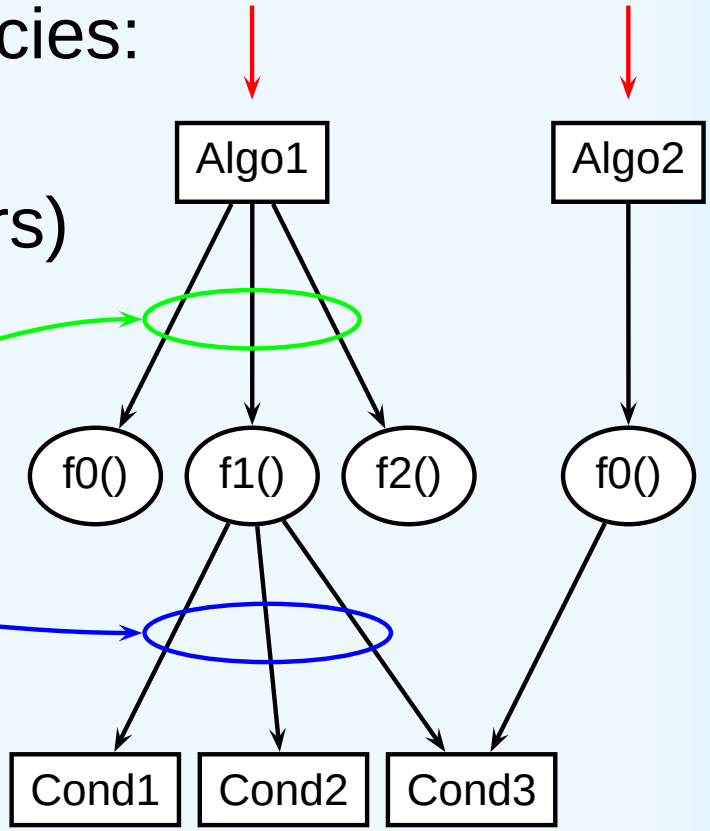
- start from the head
  (objects without consumers)
- intersection of validities
  - object level
    (methods)



Update Policy

Given a network of dependencies:

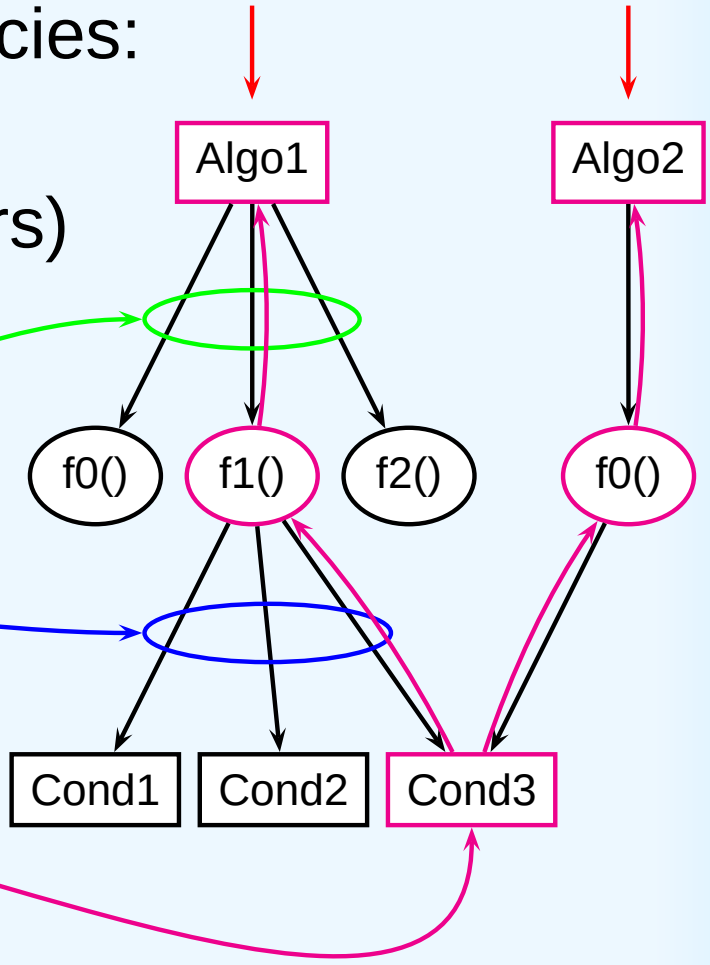
- start from the head
  (objects without consumers)
- intersection of validities
  - object level
    (methods)
  - method level
    (child objects)



Update Policy

Given a network of dependencies:

- start from the head
  (objects without consumers)
- intersection of validities
  - object level
    (methods)
  - method level
    (child objects)
- invalid object



On-Line

On-Line environment

- Local Area Network in the Pit
- $\sim$4000 processes
  - conditions loaded at initialization (alignments, trigger configurations,. . .)

On-Line environment

- Local Area Network in the Pit
- $\sim$4000 processes
  - conditions loaded at initialization
    (alignments, trigger configurations,. . .)

$\Rightarrow$ impossible to use directly a database server

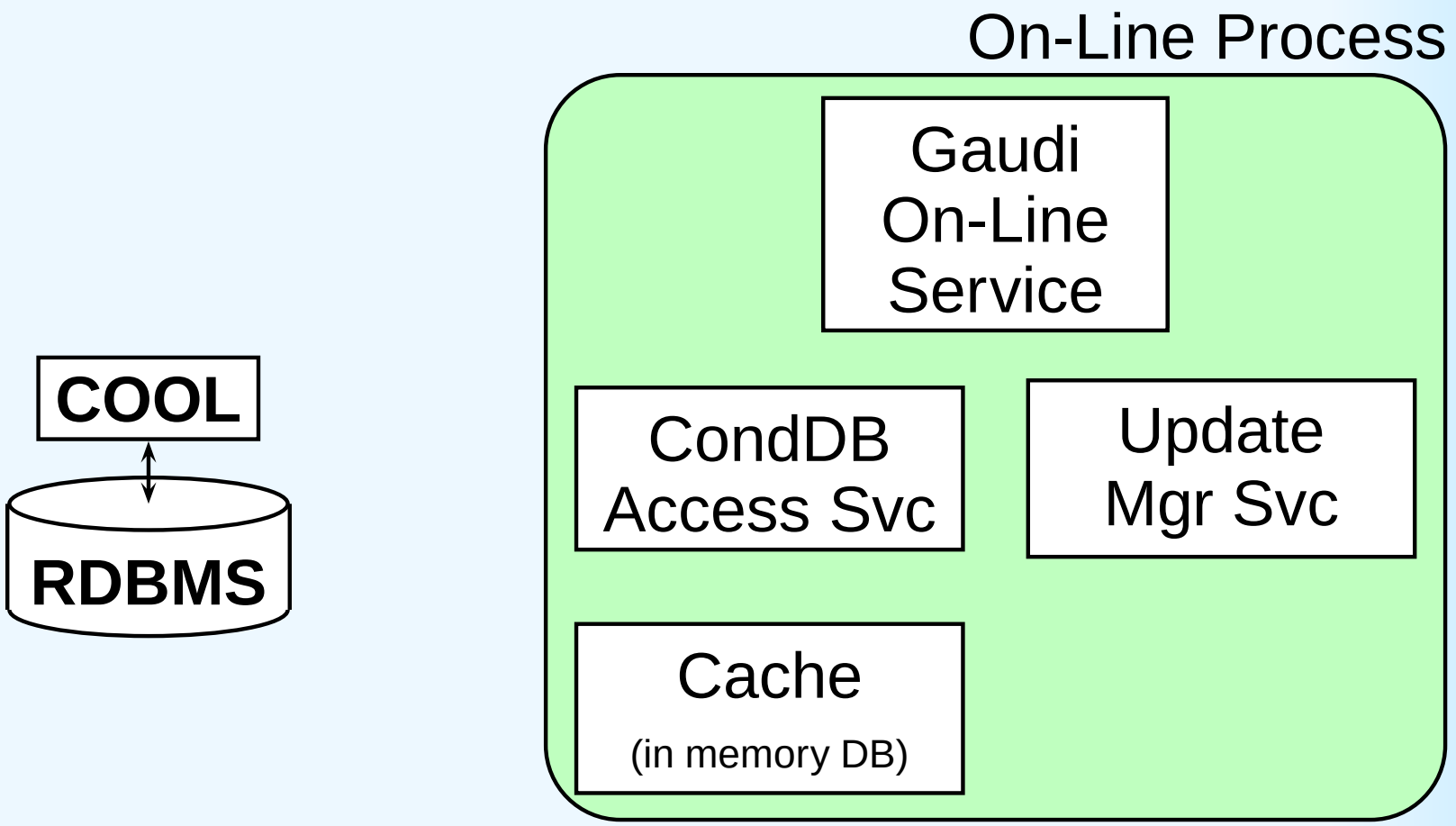
On-Line environment

- Local Area Network in the Pit
- $\sim$4000 processes
  - conditions loaded at initialization
    (alignments, trigger configurations,. . .)

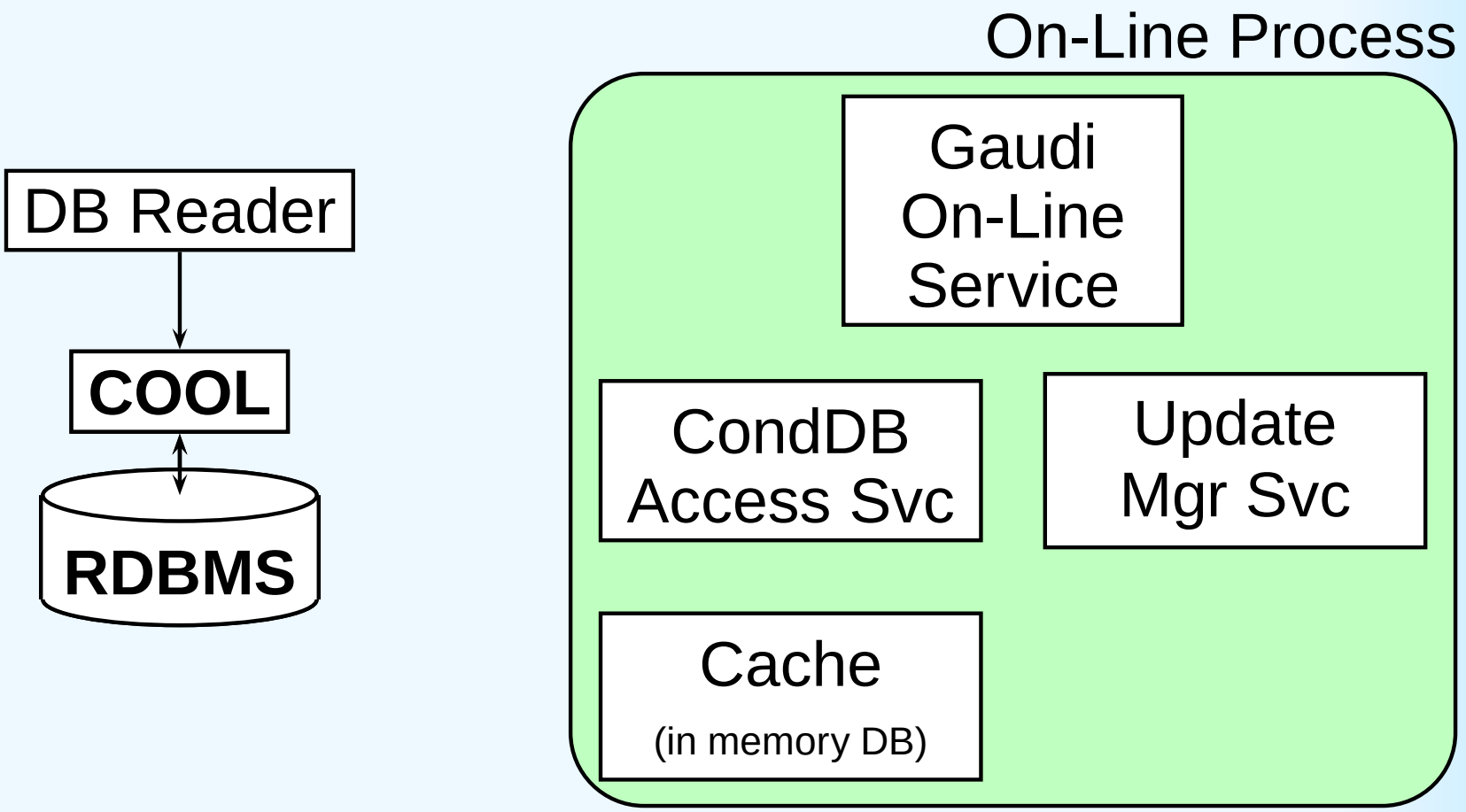
$\Rightarrow$ impossible to use directly a database server

we can use the cache of the
CondDB Access Service

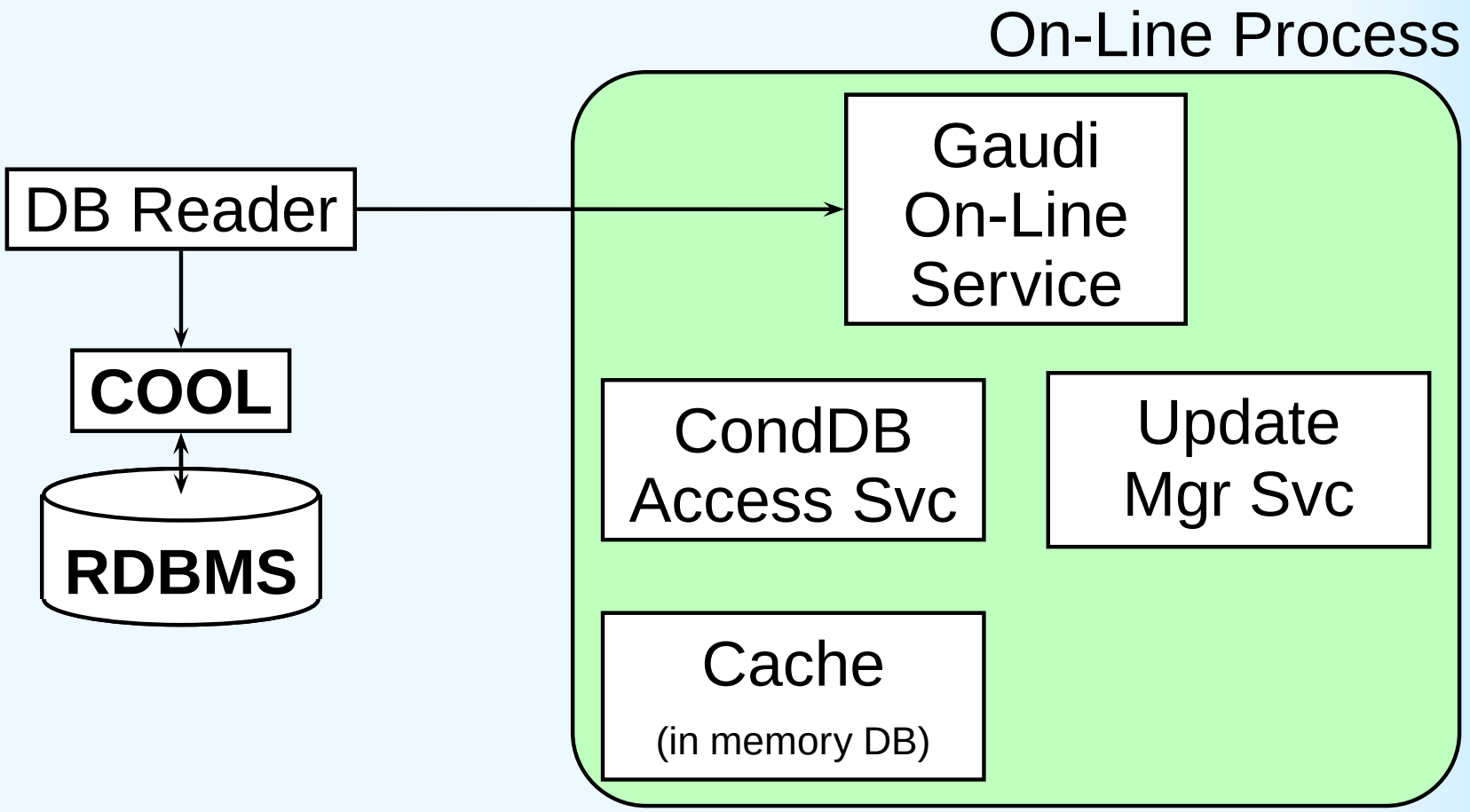
CondDB and On-Line: Initialization



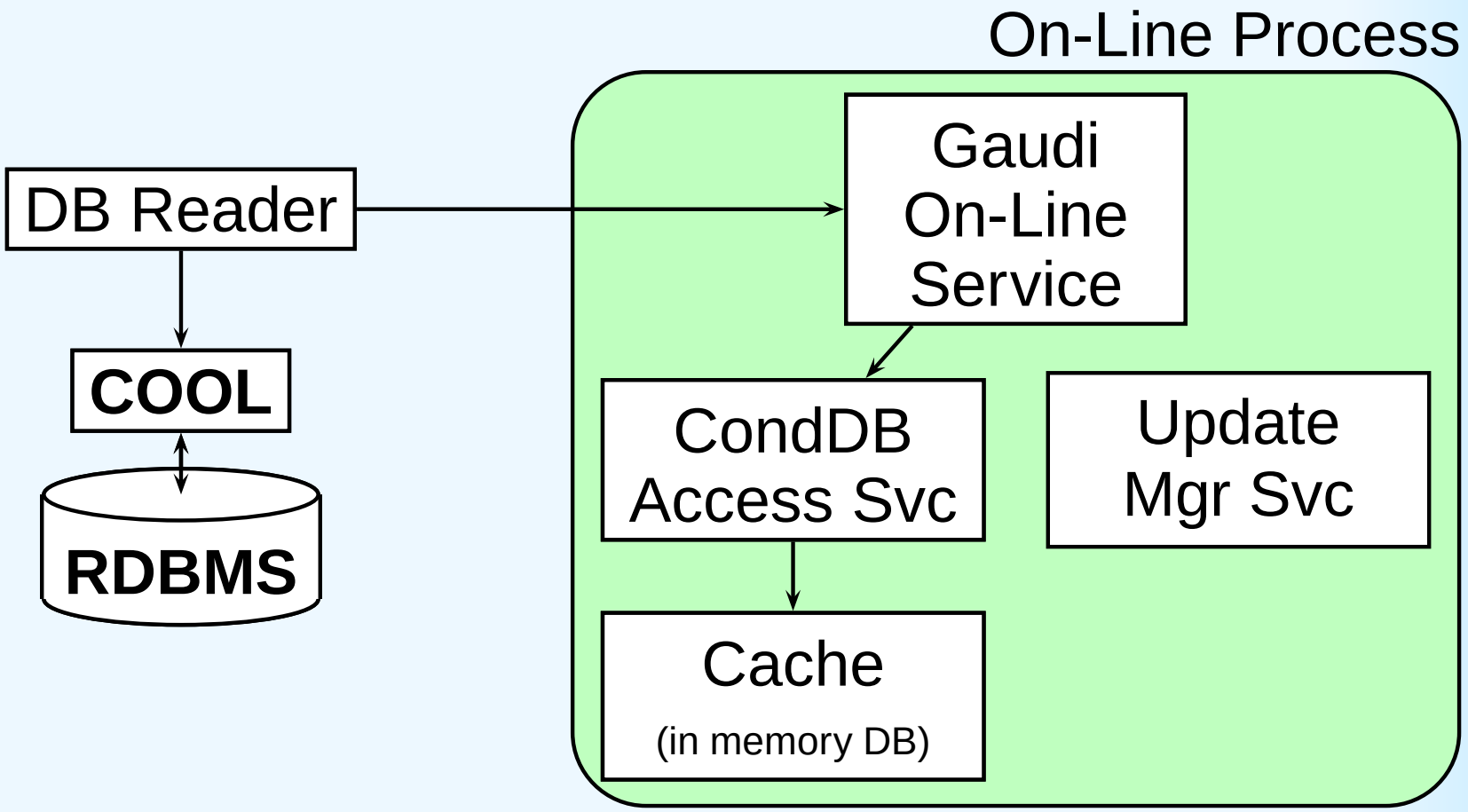
CondDB and On-Line: Initialization



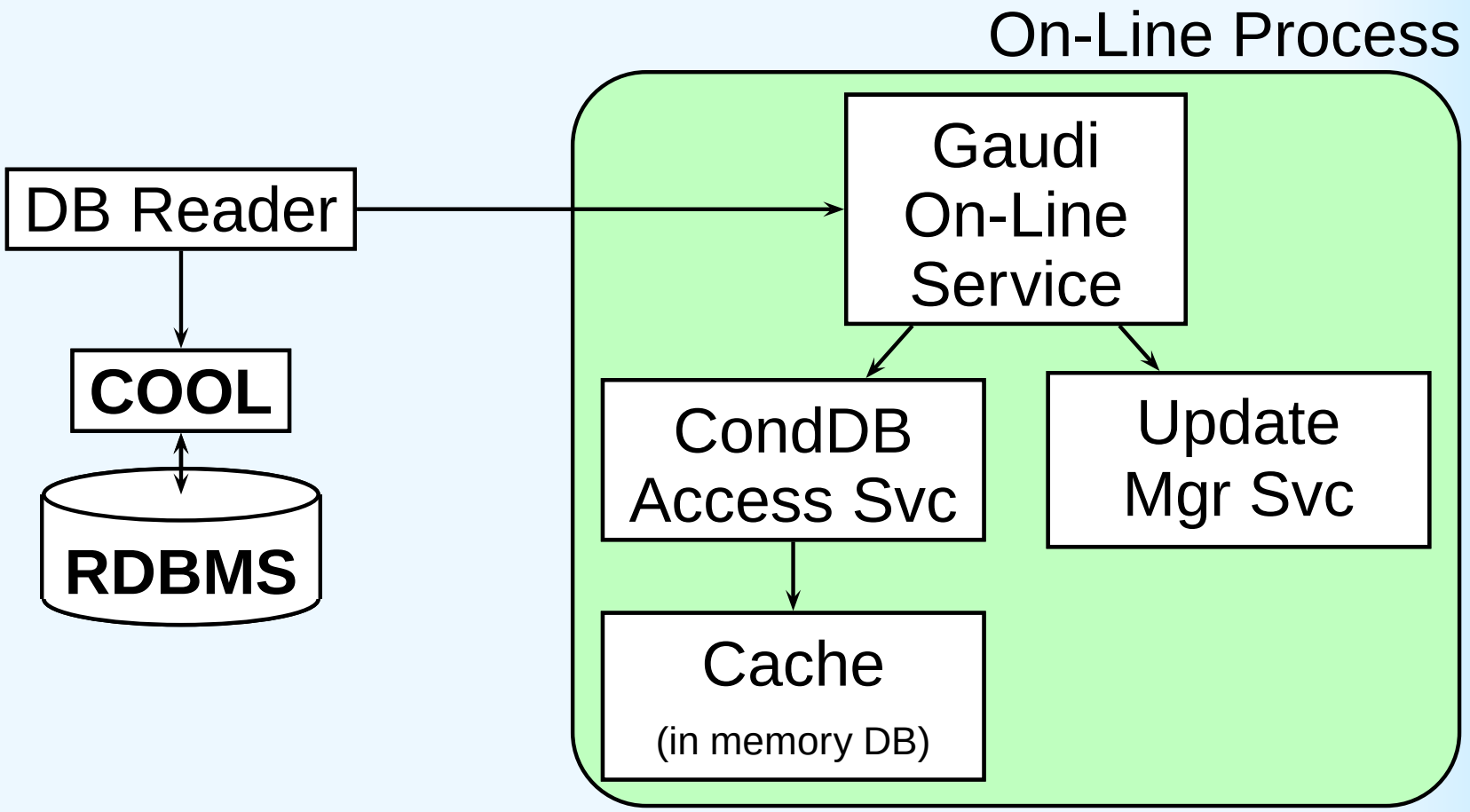
CondDB and On-Line: Initialization



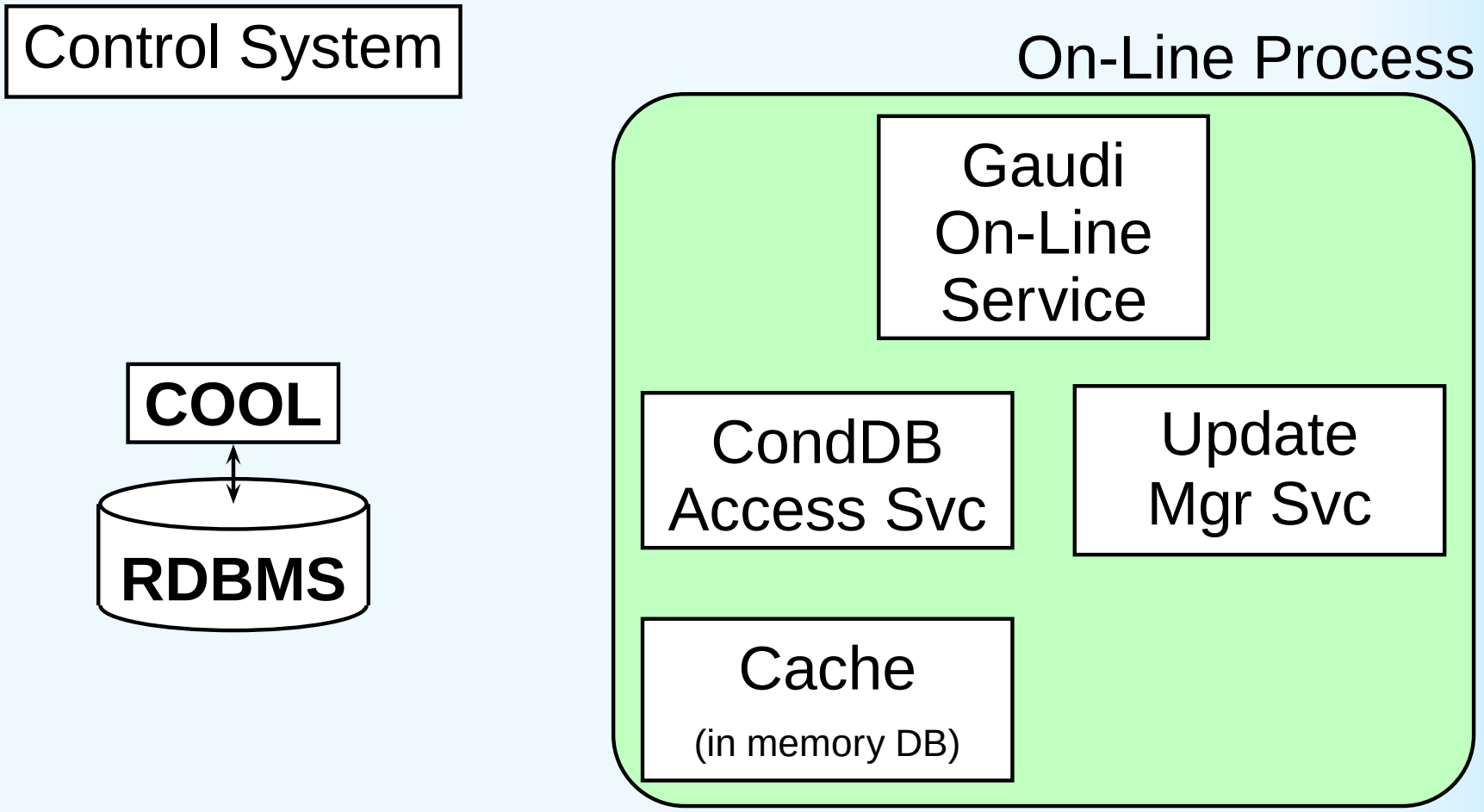
CondDB and On-Line: Initialization



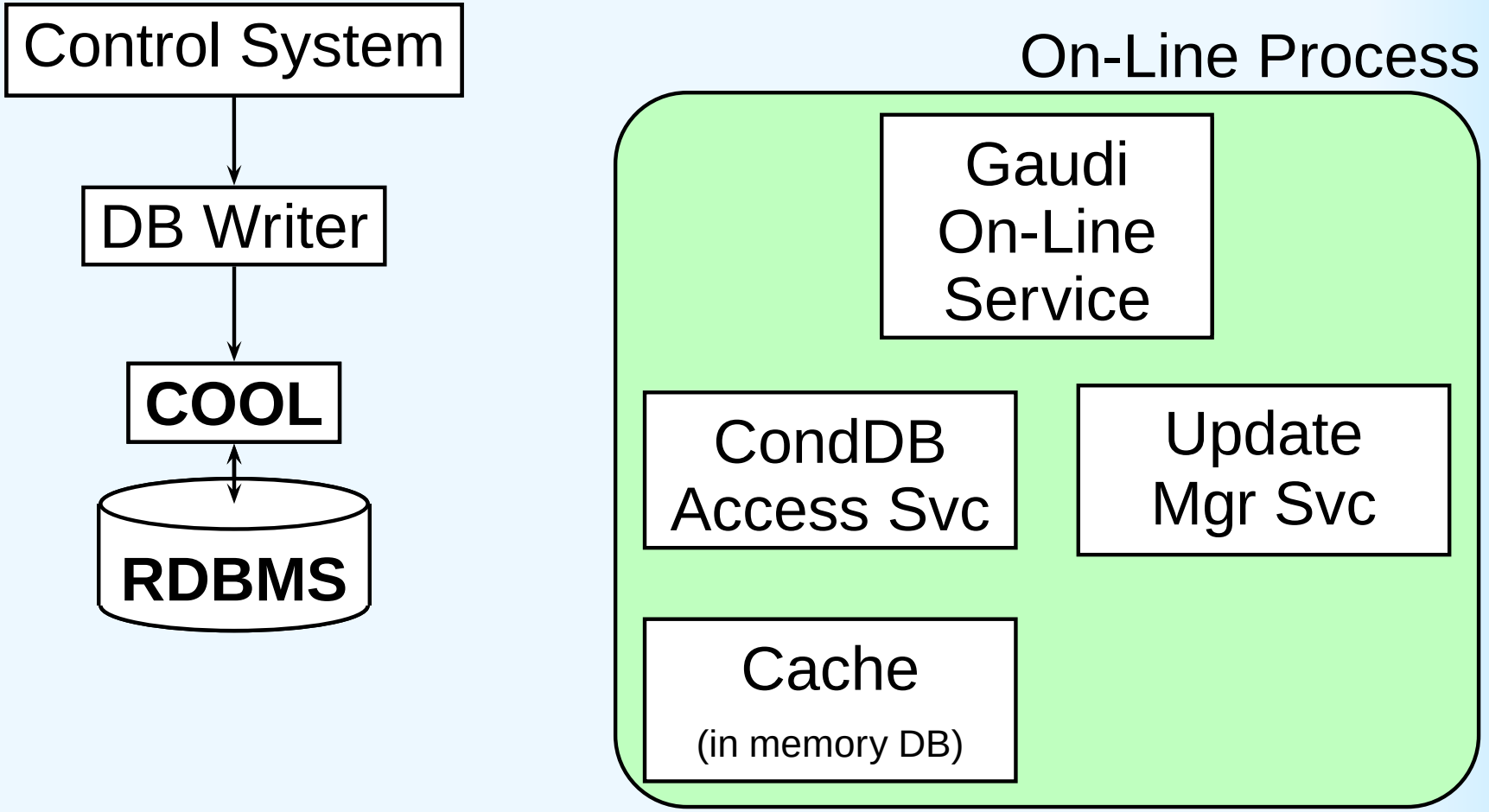
CondDB and On-Line: Initialization



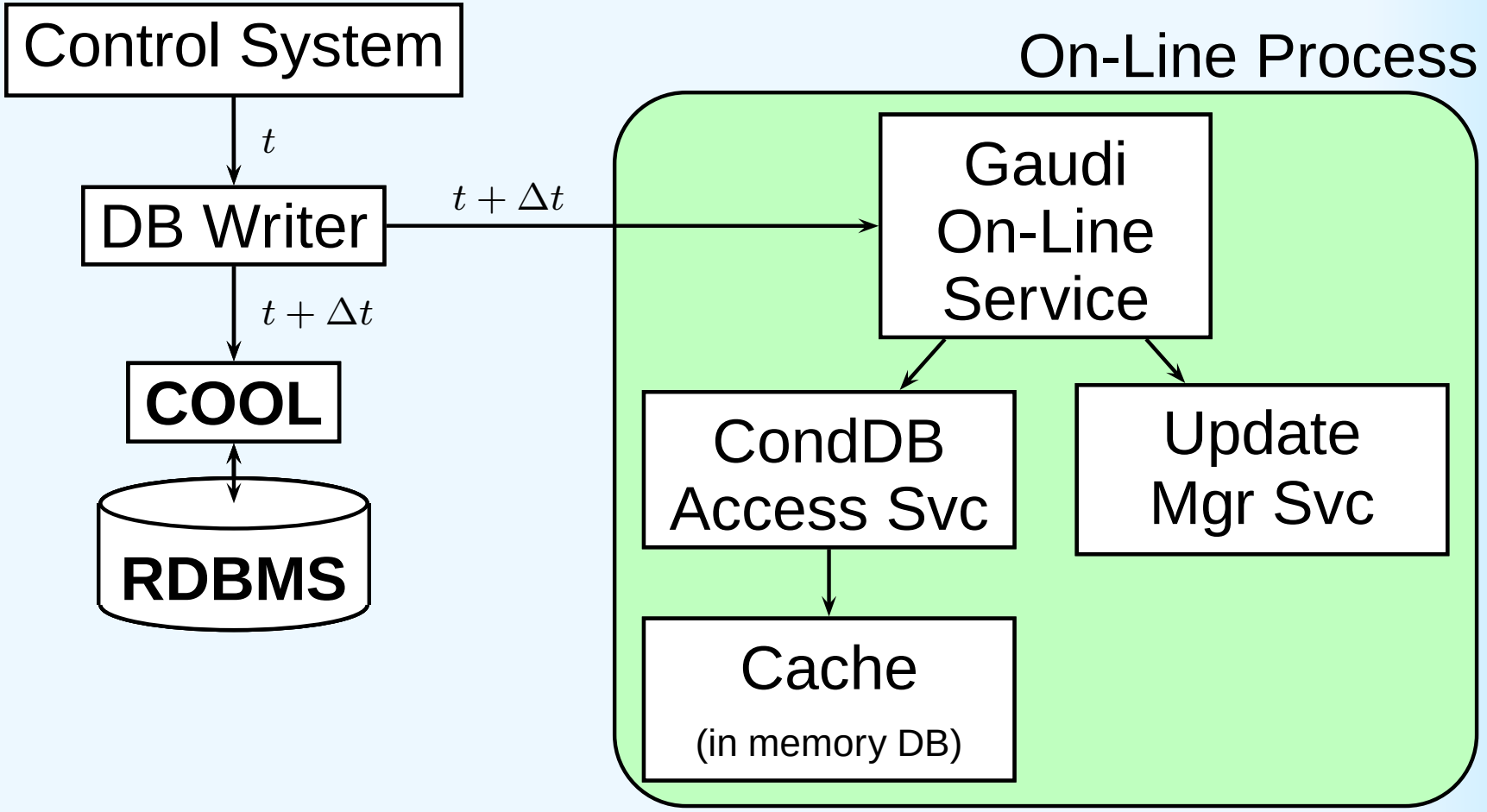
CondDB and On-Line: Run



CondDB and On-Line: Run

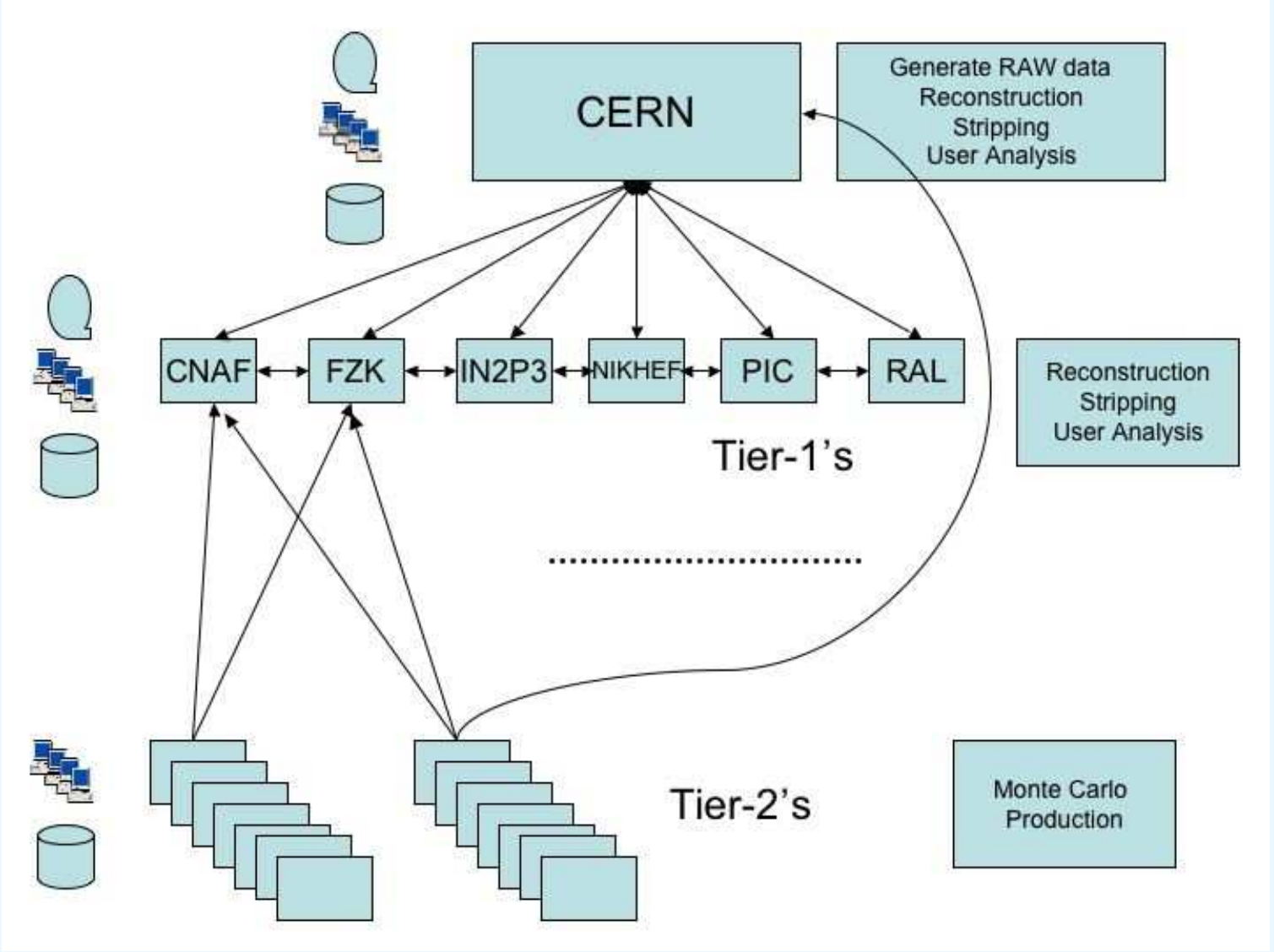


CondDB and On-Line: Run

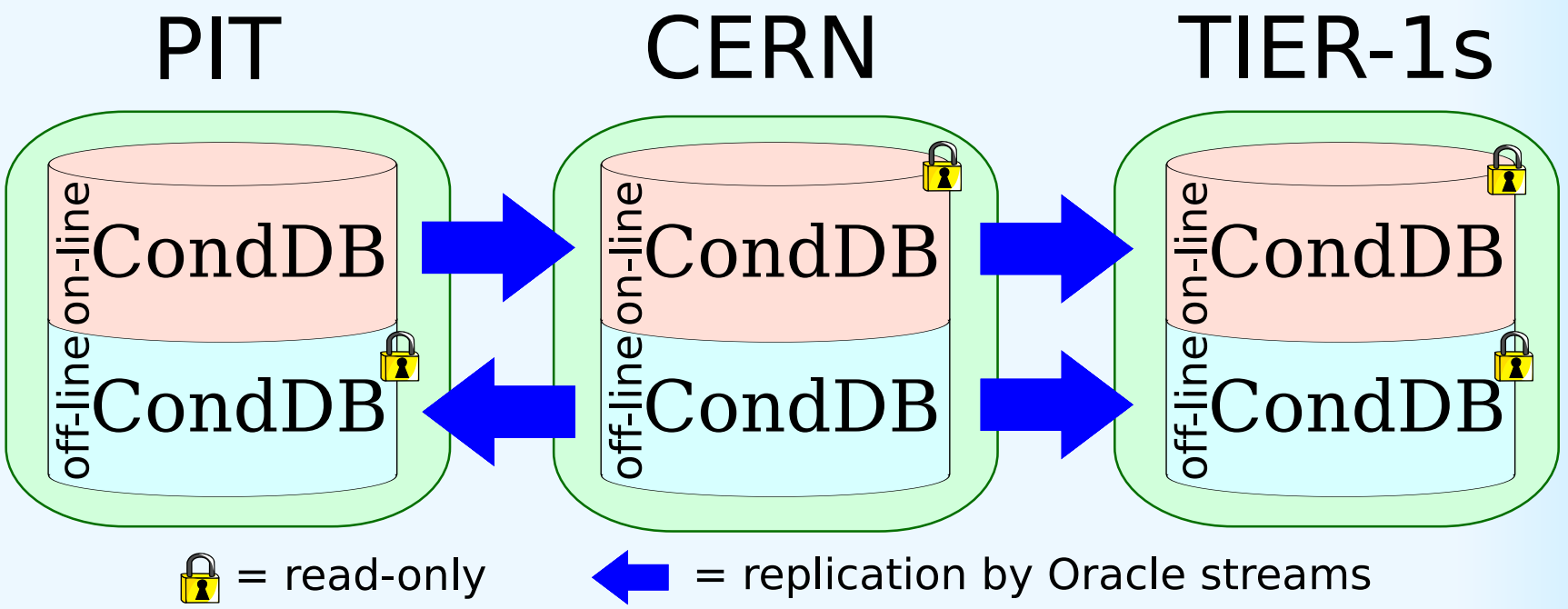


Deployment

Computing Model



Deployment Plan



- Master copies at PIT and CERN (synchronized)
- Copies at Tier-1s
- Expected DB size: few GB

Summary

Summary

- LHCb Conditions Database
  - LGC project COOL for the storage
  - integrated in the Gaudi persistency framework

Summary

- LHCb Conditions Database
  - LGC project COOL for the storage
  - integrated in the Gaudi persistency framework
- Update Mechanism
  - flexible
  - simple to use
  - optimized

Summary

- LHCb Conditions Database
  - LGC project COOL for the storage
  - integrated in the Gaudi persistency framework
- Update Mechanism
  - flexible
  - simple to use
  - optimized
- Online Usage
  - system to publish conditions to the on-line farm