

Status of MCDB. Automatic Uploading Interface

L. Dudko, SINP MSU
on behalf of LCG MCDB group

<http://mcdb.cern.ch>

LCG MCDB group:

S. Belov, JINR

L. Dudko, SINP MSU

A. Ribon, CERN

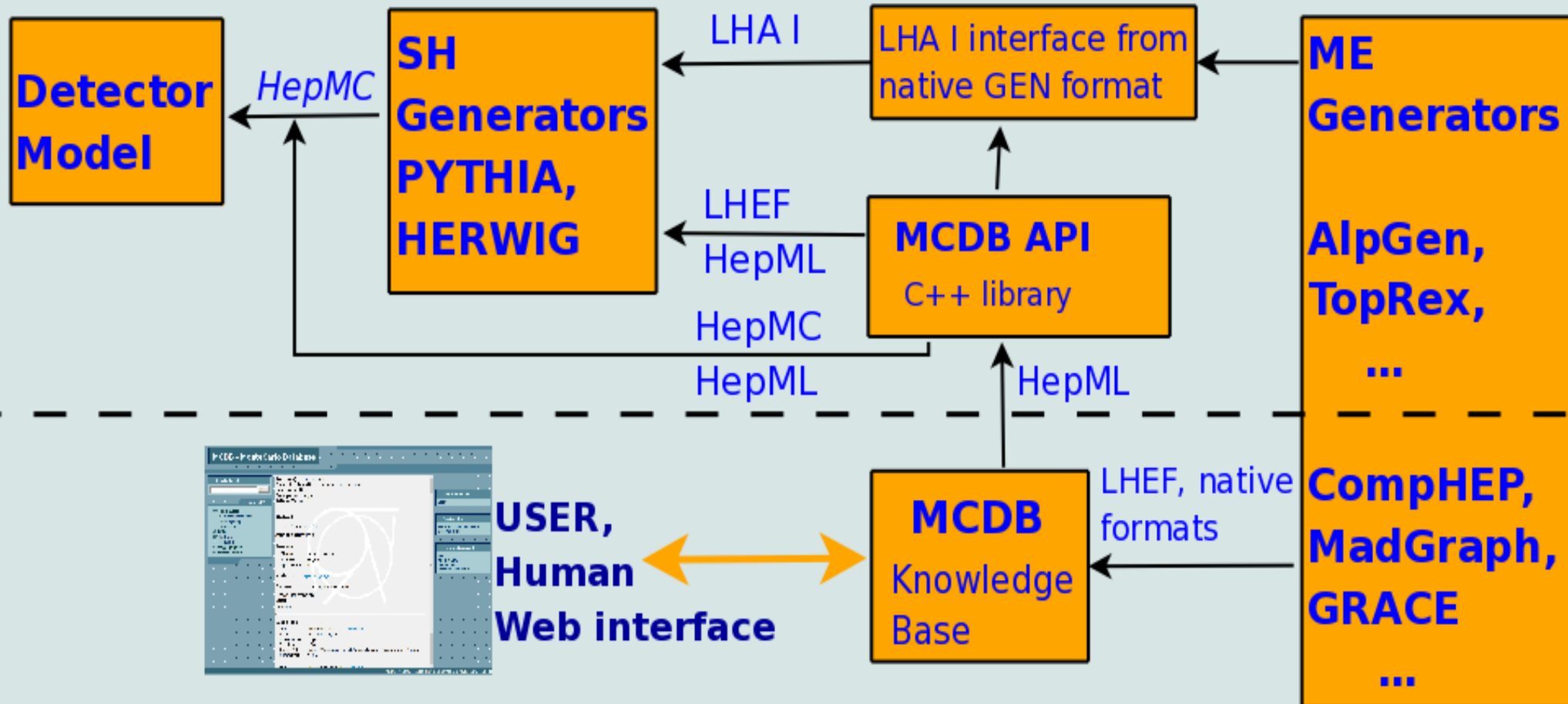
*A. Sherstnev, Univ. of
Cambridge*

OUTLINE:

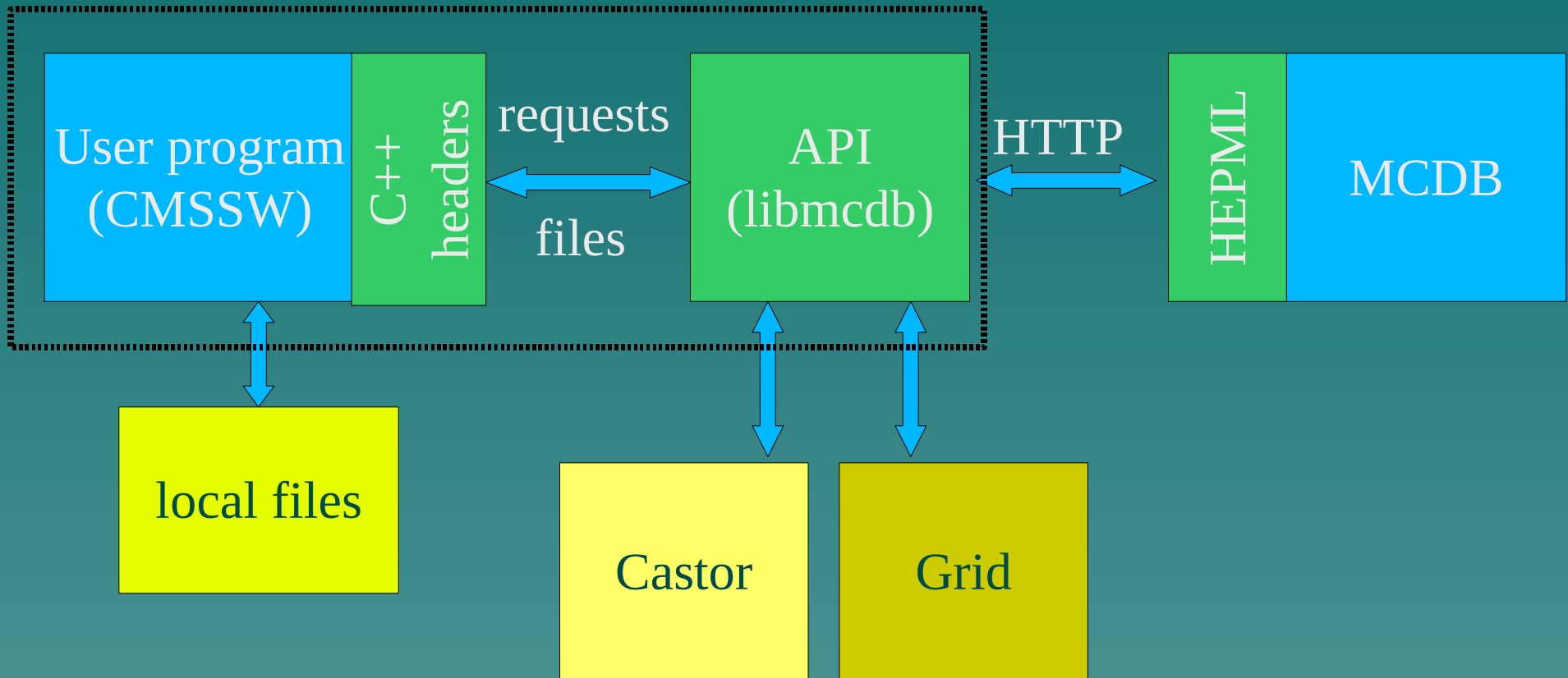
- MCDB interfaces
- Automatic uploading of the samples to MCDB

MCDB in the Simulation Chain

Collaboration Software Simulation Chain



MCDB API structure



Automatic access to the samples from MCDB in CMSSW production chain

LibMCDB realisations

(LibMCDB / MCDB API / MCDBInterface)

- ◆ Standalone package for everybody
(available on MCDB web site)

- ◆ Integrated in CMSSW as additional
common interface

(Unknown official status in CMSSW CVS?)

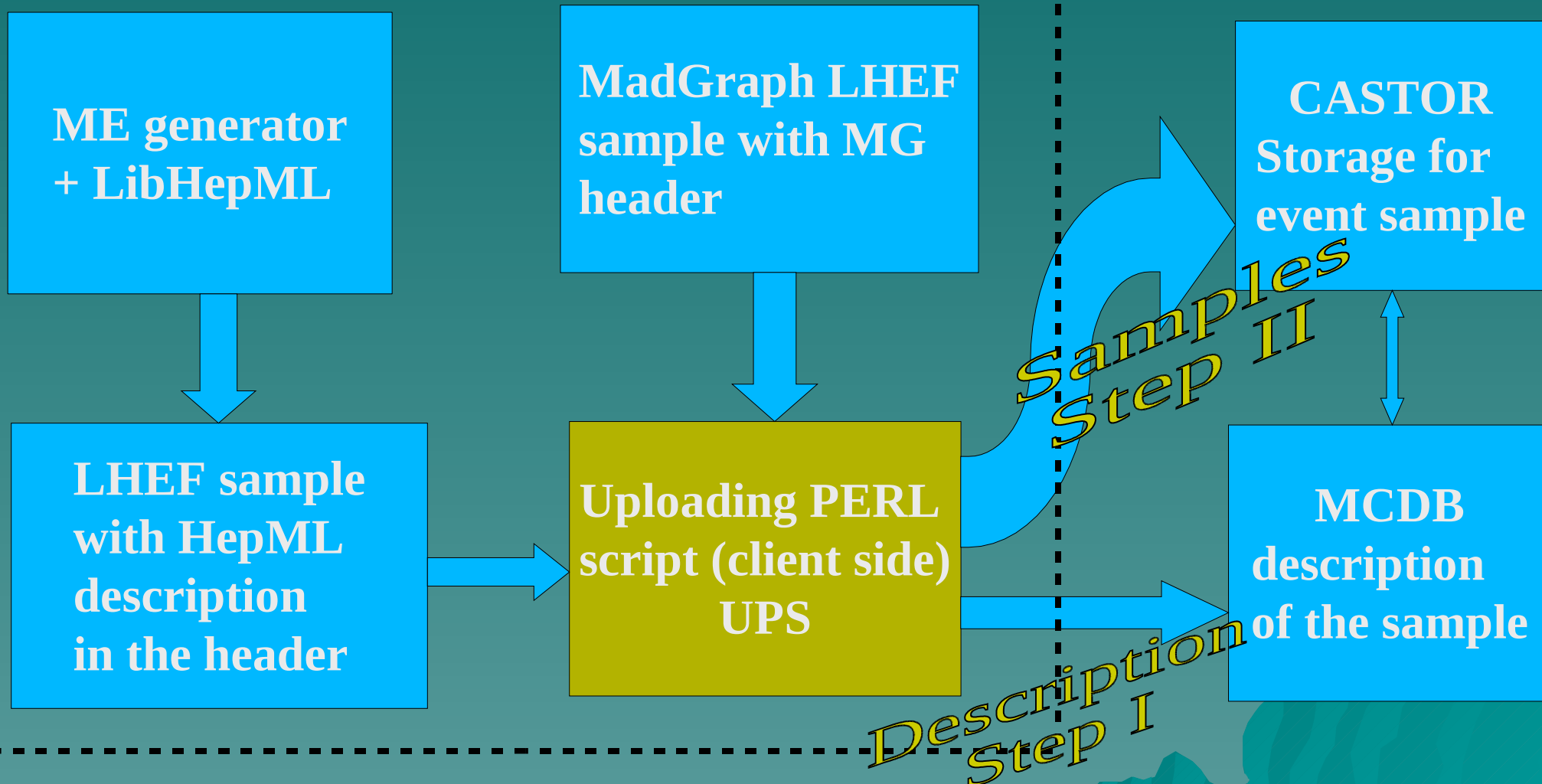
CMSSW/src/GeneratorInterface/MCDBInterface/

- ◆ External LCG AA package
in /afs/cern.ch/sw/lcg/external/mcdb/

We support all of these realisations for a while.

Automatic Uploading to MCDB

Client Side



Step I: Uploading Script (UPS) \Leftrightarrow MCDB

- ◆ UPS parses LHEF to find the header block
- ◆ UPS authorizes in MCDB and pass LHEF header to MCDB via HTTP (POST)
- ◆ MCDB parses the LHEF header (HepML or MadGrpah) to find the specific elements of sample description (According to SQL structure)
- ◆ MCDB creates new article and specific directory on CASTOR
- ◆ MCDB returns to UPS ArticleID and CASTOR directory to upload the sample

Step II: Uploading Script (UPS) => CASTOR

- ◆ UPS Authorizes on CASTOR with Grid certificates: `grid-proxy-init`
- ◆ UPS uploads the sample(s) to the specific incoming CASTOR directory via `globus-url-copy`

Possible Uploading Issue:

- CASTOR team plans to close Castorgrid classic Storage Element and provide only Storage Resource Manager (SRM) access.

CASTOR team continue to support `globus-url-copy` and `lcg-cp`

Present path: `gsiftp://castorgrid.cern.ch/castor/cern.ch/sft/mcdb/incoming/`

We investigate this issue.

Authorization Mechanisms

- ◆ UPS => MCDB authorization
(the corresponding login or DN should be registered in MCDB first)
 - CERN AFS login/password for UPS
 - LCG PKCS12 certificate + password
 - LCG usercert.pem/userkey.pem from ~/.globus/ require password
 - grid-proxy-init (default; no password)
- ◆ UPS => CASTOR authorization
 - grid-proxy-init (first check with grid-proxy-info)

Uploading Script Overview

- ◆ One LHEF header sample description is one article in MCDB
- ◆ UPS takes samples description from the first file and creates MCDB article
- ◆ All other files are simple copied to CASTOR and attached to the same article
- ◆ Different physics processes should be uploaded by different runs of UPS
- ◆ In the first version of UPS only MadGraph header is possible, HepML is in progress

HOW TO Use UPS, Task I

The main task is to describe the set of LHEF (MadGraph or HepML header) files in MCDB as the new article and upload the files to specific CASTOR directory.

./upload2mcdb.pl file1 file2 ...

Description for the new MCDB article will be taken from the header of file1 but all other files will be uploaded and attached to the same article.

Possible additional options:

- dsname DataSetName #specify Data Set Name
- header [MG, hepml] #specify type of LHEF header (MG - MadGraph, hepml - HepML header)
- authors AFSlogin1,AFSlogin2,... #set additional authors for the article
- category Category1,Category2,... #set MCDB Category where to attache article (default is CMS08MG)
- not2web #do not post Article to WEB (keep in MCDB), default is post right after it is described
- verbose #be verbose during the run
- debug #print additional information during the session

HOW TO Use UPS, Task II

- Upload more samples to the existing MCDB article
- Requires ArticleID or DataSetName to identify where to attach the samples
- Do not change the description
- Not only LHEF format is possible

```
upload2mcdb.pl [-artid N] [-dsname DataSetName] --uploadonly file1 file2 ...
```

Possible options:

- verbose #be verbose during the run
- debug #print additional information during the session

HOW TO Use UPS, Task III

Replace the description in the existing article and upload new files

```
upload2mcdb.pl [-artid N] [-dsname DataSetName] --replace file1 file2 ...
```

Possible options:

- dsname DataSetName #specify Data Set Name (analog ArticleID)
- header [MG or hepml] #specify type of LHEF header (MG - MadGraph, hepml - HepML header)
- authors AFSlogin1,AFSlogin2,... #set additional authors for the article
- category Category1,Category2,... #set MCDB Category where to attache article (default is CMS08MG)
- not2web #do not post Article to WEB (keep in MCDB), default is post right after it is described
- verbose #be verbose during the run
- debug #print additional information during the session

HOW TO Use UPS, Task IV

Describe sample in new article but do not upload any file to CASTOR.
The description is taken from the header of the file

upload2mcdb.pl --descriptiononly file1

Possible additional options:

- | | |
|-----------------------------------|---|
| -dsname DataSetName | #specify Data Set Name |
| -a [login, pkcs12, cert, globus] | #type of authorization in MCDB, default is globus |
| -header [MG, hepml] | #specify type of LHEF header (MG - MadGraph, hepml - HepML header) |
| -authors AFSlogin1,AFSlogin2,... | #set additional authors for the article |
| -category Category1,Category2,... | #set MCDB Category where to attach article (default is CMS08MG) |
| -not2web | #do not post Article to WEB (keep in MCDB), default is post right after it is described |
| -verbose | #be verbose during the run |
| -debug | #print additional information during the session |

HOW TO Get Help on UPS

Short help. Get the list of available options:

```
./upload2mcdb.pl -h
```

Long help. Get the detailed description with examples:

```
./upload2mcdb.pl --help
```

Where to Find MCDB Uploading Interface Scripts

Client Part of Uploading Interface: *upload2mcdb.pl*

CVS: <http://simu.cvs.cern.ch/cgi-bin/simu.cgi/simu/GENSER/MCDB/distribution/>

Stable Version: <http://mcdb.cern.ch/distribution/>

Server Part of Uploading Interface: *upload_server.cgi*

CVS: <http://simu.cvs.cern.ch/cgi-bin/simu.cgi/simu/GENSER/MCDB/cgi-bin/authors/>

Summary

- ◆ Three MCDB interfaces are ready:
 - Interactive WEB interface (everything is possible)
 - MCDB API is the automatic way to process the event samples from MCDB during the production (not only LHEF samples)
 - Uploading Interface is the automatic way to upload and describe LHEF sample(s) to MCDB (First version works with MG header only)
- ◆ LibHepML is the proposed standard way to describe the sample automatically in ME generator (see dedicated talk by S.Belov, LCG meeting 9.04.08)

MCDB PLANS

- ◆ Long TO-DO list for different parts of MCDB (available in CVS)
 - WEB Interface
 - Storage System
 - Physics data representation
 - Help and support
 - MCDB Software
- ◆ Improve Uploading Interface
- ◆ Continue integration with CMSSW software
- ◆ Reorganize MCDB software and publish it as an Open Source Content Management System

BACKUP SLIDES

Overview of the Knowledge Base

- **The Major Tasks of LCG MCDB**
 - **Share sophisticated MC generated samples between different groups**
 - **Samples prepared by experts in MC event generation**
 - **Resource-intensive samples (Human or/and CPU resources)**
 - **Provide infrastructure to keep MC samples and sample documentation**
 - **Facilitate communication between MC experts and users in LHC collaborations**

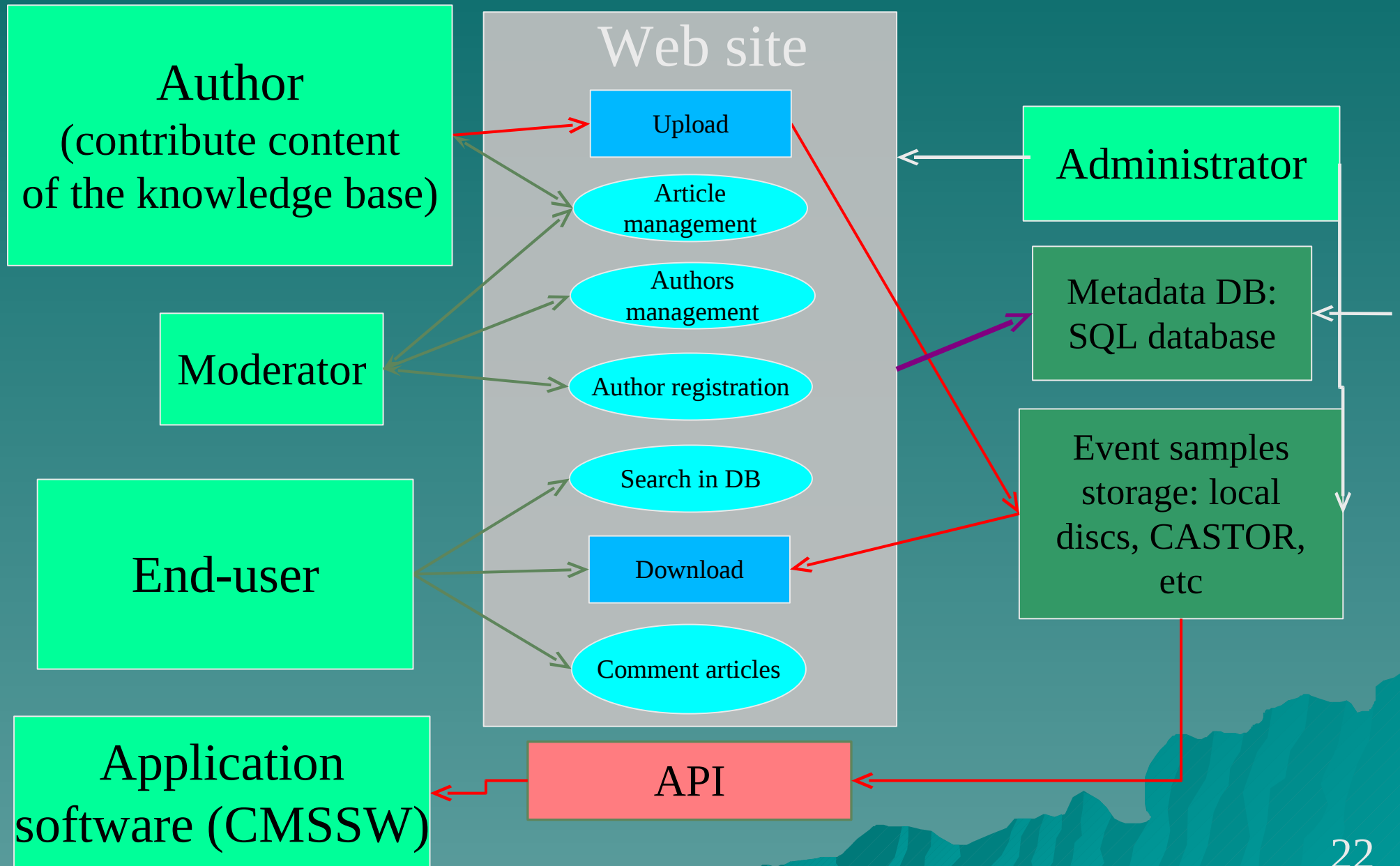
The Major Features of LCG MCDB (I)

- ◆ Powerful WEB interface with Content Management System for the authors of event samples and end-users
- ◆ Tree graph of physics categories with articles published in MCDB to browse the database content
- ◆ Power search engine based on SQL/XML to search the content of the knowledge base
- ◆ Flexible and reliable authorization system based on CERN AFS/Kerberos logins or LCG GRID certificates
- ◆ SQL structure of event sample documentation
- ◆ BackUp of samples and SQL information

The Major Features of LCG MCDB (II)

- ◆ CASTOR is the native storage for the event samples
- ◆ Direct uploading of multiple files from AFS/CASTOR/GRID (wild-card characters are possible) to LCG MCDB
- ◆ Direct downloading of files from LCG MCDB (CASTOR) with HTTP/RFIO/GridFTP/... (URI)
- ◆ Application Programming Interface (API) for the LHC collaborations environment software
- ◆ LHEF/HepML unification of event file format and sample description

LCG MCDBB Scheme



PARAMETERS OF EVENT SAMPLE DESCRIPTION

MCDB XML Scheme inside HepML specifications

◆ General information

- Title
- Abstract
- Authors
- Experiment and/or Group

◆ Physics process

- Initial state
- Final state
- QCD scale
- Process PDF

◆ Event files

- Physics process/subprocesses
- File name
- Events number
- cross section and uncertainty

◆ Used generator

- Name and version
- Description
- Home page address

◆ Theoretical model

- Name
- Description
- Set of parameters and their values with author's descriptions

◆ Applied cuts

STEP I: MCDB \Leftrightarrow API

- ◆ HTTP request with ArticleID from API to MCDB server
(e.g. <http://mcdb.cern.ch/cgi-bin/xmlquery.cgi?article=116>)
- ◆ HepML description and paths to samples, as an answer from MCDB to API
- ◆ API parses the HepML (XML) block and fill C++ classes with the description
- ◆ API make local copy of the event sample, download the remote sample via RFIO, GridFTP, HTTP, ...

STEP II: API => CMSSW

- ◆ API provides to CMSSW:
 - local path to file with input events
 - Sample Description in the form of C++ objects described in `mcdb.hpp`
- ◆ CMSSW process the local event file to the next level of simulation
- ◆ CMSSW passes the C++ objects with event description to the output files

MCDB API C++ Classes

<http://mcdb.cern.ch/doc/API/public/mcdb.hpp>

```
namespace mcdb
{
class MCDB;
class Article;
class File;
class Author;
class Cut;
class Generator;
class Model;
class Process;
class Subprocess;
}
```

```
class Generator{
public:
    Generator();
    ~Generator();
    string& name();
    string& name(const string&);
    string& version();
    string& version(const string&);
    string& homepage();
    string& homepage(const string&);
class Process{
public:
    Process();
    ~Process();
    string& initialState();
    string& initialState(const string&);
    string& finalState();
    string& finalState(const string&);
    string& factScale();
    string& factScale(const string&);
    string& renormScale();
    string& renormScale(const string&);
    string& pdf();
    string& pdf(const string&);
```

```
class Model{
public:
    Model();
    ~Model();
    class ModelParameter;
    string& name();
    string& name(const string&);
    string& description();
    string& description(const string&);
    vector<ModelParameter>& parameters();
    vector<ModelParameter>&
    parameters(const vector<ModelParameter>&);
class ModelParameter
{
public:
    ModelParameter();
    ~ModelParameter();
    string& name();
    string& name(const string&);
    string& value();
    string& value(const string&);
};
```

How it works in development CMSSW version

MCDB API library: CMSSW/src/GeneratorInterface/MCDBInterface/

BuildFile in any other interface (CompHEP, MadGraph, AlpGen, ...):

```
<use name=GeneratorInterface/MCDBInterface>
```

Inside any other interface (e.g. src/GeneratorInterface/ComphepInterface/src/ComphepSource.cc):

```
#include "GeneratorInterface/MCDBInterface/interface/MCDBInterface.h"
```

```
mcdb::Article SampleDescription;
```

```
if (getInputFromMCDB_) {
```

```
    CHFile_ = pset.getUntrackedParameter<string>("ComphepInputFile");
```

```
    mcdbGetInputFile(CHFile_, MCDBArticleID_, SampleDescription);
```

```
}
```

Config: part of src/GeneratorInterface/ComphepInterface/test/testMCDBInterface.cfg:

```
untracked bool getInputFromMCDB = true
```

```
int32      MCDBArticleID = 118
```

```
string ComphepInputFile =
```

```
"rfio://castor/cern.ch/sft/mcdb/storage/118/wbbj_mu.pev"
```

```
#string ComphepInputFile = "wbbj_mu.pev"
```

#Can be any of SEAL protocols (RFIO, GridFTP, HTTP, ...) from any locations

Automatically process and document sample from MCDB in CMSSW

History: CMS MCDB

- <http://cmsdoc.cern.ch/cms/generators/mcdb/>
- Operated in CMS during the OSCAR/ORCA era, widely used by the Higgs group
- Only parton level files; AFS storage; Only phonetic search; No SQL

The screenshot shows a web browser window titled "LCG Monte-Carlo Events DataBase - Mozilla". The page content includes:

- Header:** "Monte-Carlo Events Data Base" with CERN and CMS logos.
- Left Sidebar:** A navigation menu with links: HIGGS, TOP, W and n jets, Z and n jets, Gamma and n jets, WW and n jets, ZZ and n jets, WZ and n jets, Gamma Gamma n jets, W Gamma n jets, Z Gamma n jets, QCD multijets, REQUESTS, PROGRAMS, and FAQ.
- Main Content Area:** A list of event data entries, each with a title, description, and metadata. Visible entries include:
 - QCD 2TAU+3J EVENTS WITH ALPGEN2. CAN BE USED FOR MLM ME+PS**: QCD 2tau+3j events generated with ALPGEN2 by Maiko Takahashi. Can be used for MLM ME+PS procedure, since generated with ickw=1. published: 06/06/2005 | author: Alexandre Nikitenko | category: Z and n jets
 - QCD 2TAU+2J EVENTS WITH ALPGEN2. CAN BE USED FOR MLM ME+PS**: QCD 2tau+2j events generated with ALPGEN2 by Maiko Takahashi. Can be used for MLM ME+PS procedure, since generated with ickw=1. published: 06/06/2005 | author: Alexandre Nikitenko | category: Z Gamma n jets
 - LO gg->W*W*->2L EVENTS, L=E, MU, TAU**: LO gg->W*W*->2l events provided by Nikolas Kauer for gg->H->WW*->2l study during Les Houches 2005 Workshop. The information about generator can be found on Higgs group page. published: 19/05/2005 | author: Alexandre Nikitenko | category: WW and n jets
 - PHOTON + 3 JETS, QCD DIAGRAMS, COMPLETE TREE LEVEL SETS, COMPHEP, 850K EVENTS**: QCD fake background to the light Higgs signal in the W,Z fusion (gamma gamma + 2 jets channel). 850K event sample generated by CompHEP 4.2p1. published: 25/04/2005 | author: Mikhail Dubinin | category: Gamma and n jets
 - PP->tt* + GAMMA GAMMA, T1(2)->Wb->QQb, T2(1)->Wb->B L NU (L=E,MU,TAU) GENERATED BY MADGRAPH II**: pp->tt* + gamma gamma, t1(2)->Wb->qqb, t2(1)->Wb->b l nu (l=e,mu,tau) generated by Susanne Gascon with MadGraph II; gammas from ISR and FSR from top quarks. published: 25/03/2005 | author: Alexandre Nikitenko | category: TOP
 - EW TAUTAU+JJ WITH MADGRAPH. VBF AND MTAUTAU PRESELECTIONS WERE APPLIED**
- Right Sidebar:** "PUBLISH NEW DOCUMENT:" with options: non authorized author, authorized author, administrators area, and HELP.
- Bottom Left:** "SEARCH THIS SITE" with a search input field and a "search" button.

Documentation

- ◆ **Main Web Page** <http://mcdb.cern.ch>
 - ◆ Description of the project
 - ◆ Users and Authors HOW-TOs
 - ◆ Developers documentation
- ◆ **Wiki** <https://twiki.cern.ch/twiki/bin/view/LCG/LCGMCDB>
- ◆ [\[hep-ph/0404241\]](#) LCG MCDB proposal
- ◆ [\[hep-ph/0604120\]](#) LCG MCDB report (p.200-204)
- ◆ [\[hep-ph/0703287\]](#) LCG MCDB description
- ◆ **Core software supported by LCG Software Project Infrastructure** (MySQL; CASTOR; CGI; Perl; Apache)
- ◆ **Mailing lists - USERS:** lcg-mcdb-users@cern.ch
Developers: project-lcg-mcdb@cern.ch

MCDB Software

- ◆ Stable versions of MCDB server and MCDB API are available in the download section:
<http://mcdb.cern.ch/distribution>
- ◆ Development versions are in MCDB CVS:
<simu.cvs.cern.ch:/cvs/simu/GENSER/MCDB>
<http://simu.cvs.cern.ch/cgi-bin/simu.cgi/simu/GENSER/MCDB/>
- ◆ Integration of MCDB API in CMSSW CVS is in
[CMSSW/src/GeneratorInterface/MCDBInterface](#)
- ◆ All of the MCDB software will be published as an OpenSource in HepFourge

HepML in LHEF

- ◆ J. Alwall et al., A standard format for Les Houches Event Files (2006) [[hep-ph/0609017](#)] and [[hep-ph/0703287](#)]
- ◆ Event Sample Structure:

```
<LesHouchesEvents version="1.0">  
  <header>  
    <hepml>  
      <!-- HepML sample description here -->  
    </hepml>  
  </header>  
  <init> ... </init>  
  <event> ... </event>  
  <event> ... </event>  
  .....  
</LesHouchesEvents>
```

MCDB Search Engine

- *dynamic* query construction wizard (JavaScript/XML/SQL)
- Search by many possible criteria with complicated relations between DB -objects

