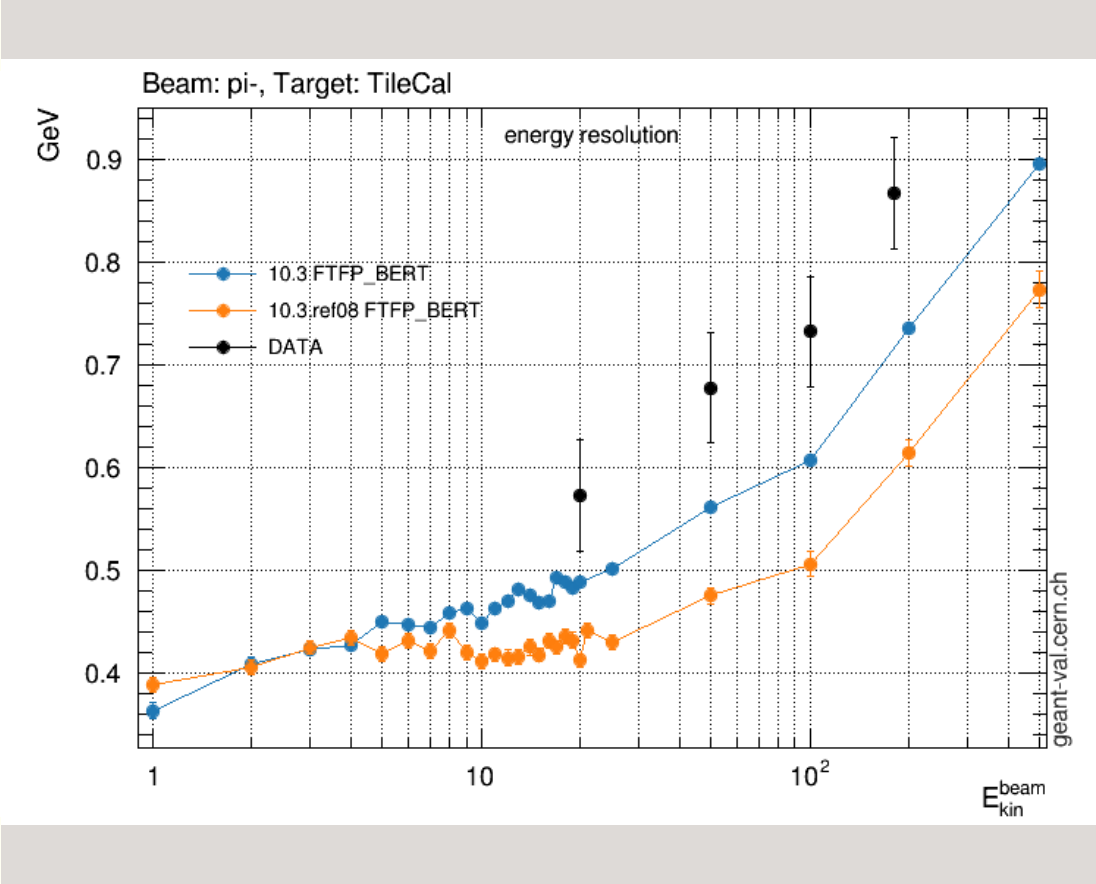
GEANT VALIDATION PORTAL:

TECHNICAL ASPECTS

Dmitri Konstantinov (IHEP, Protvino)
Grigory Latyshev (IHEP, Protvino)
Witek Pokorski (CERN)

for Geant-Val developers



Introduction

- the main purpose is to facilitate “hadronic physics” validation for Geant4
- predecessor “g4-val” was created in 2013 by George Lestaris (CERN technical student)
- the initial design was not scalable enough and slow
- In 2016 CERN summer student (*) developed a prototype based on Node.js as “interface” to DOSSIER database.

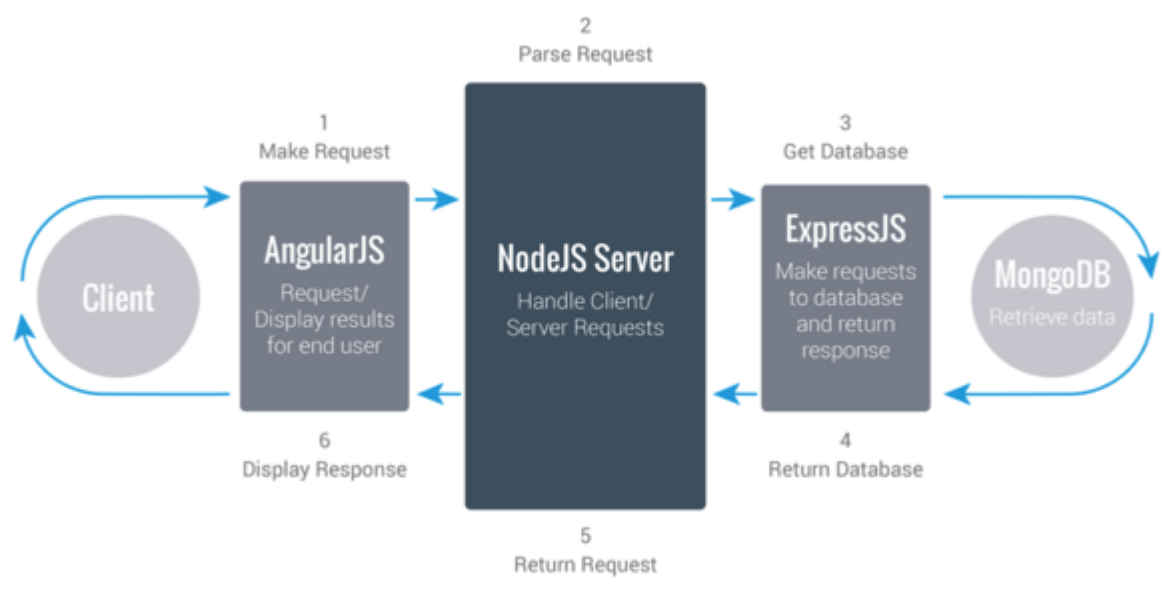
Working node.js prototype

rewrite DJANGO “g4-val” from “scratch”

(*) https://agenda.infn.it/contributionDisplay.py?contribId=113&sessionId=12&confId=11196

Webserver Architecture

- **Node.js** is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side
- Angular.js is an open-source JavaScript web framework that facilitates the creation of single-page applications and data-driven apps.
- **Express.js** is a web application framework for Node.js
- And instead of MongoDB we use PostrgreSQL



✓ Scalability
✓ Short development cycles
✓ Performance

Node.js, companies and LTS

Walmart ebay PayPal

DOW JONES intuit NETFLIX

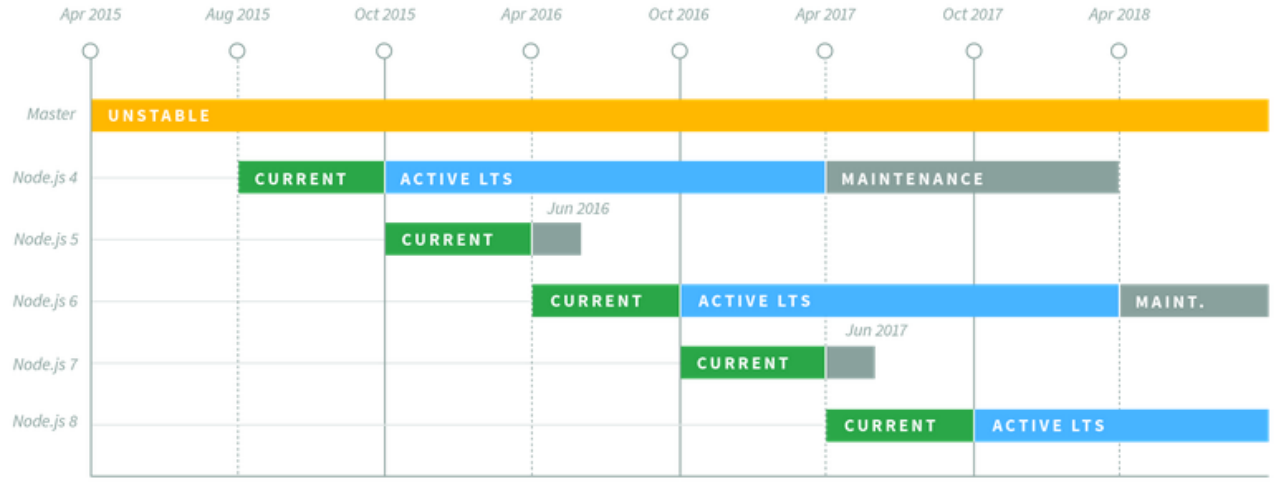
LinkedIn The New York Times Microsoft

UBER YAHOO! King

Node.js powers our web applications and has allowed our teams to move much > faster in bringing their designs to life

Jeff Harrell – Director of Engineering at PayPal

Node.js Long Term Support (LTS) Release Schedule



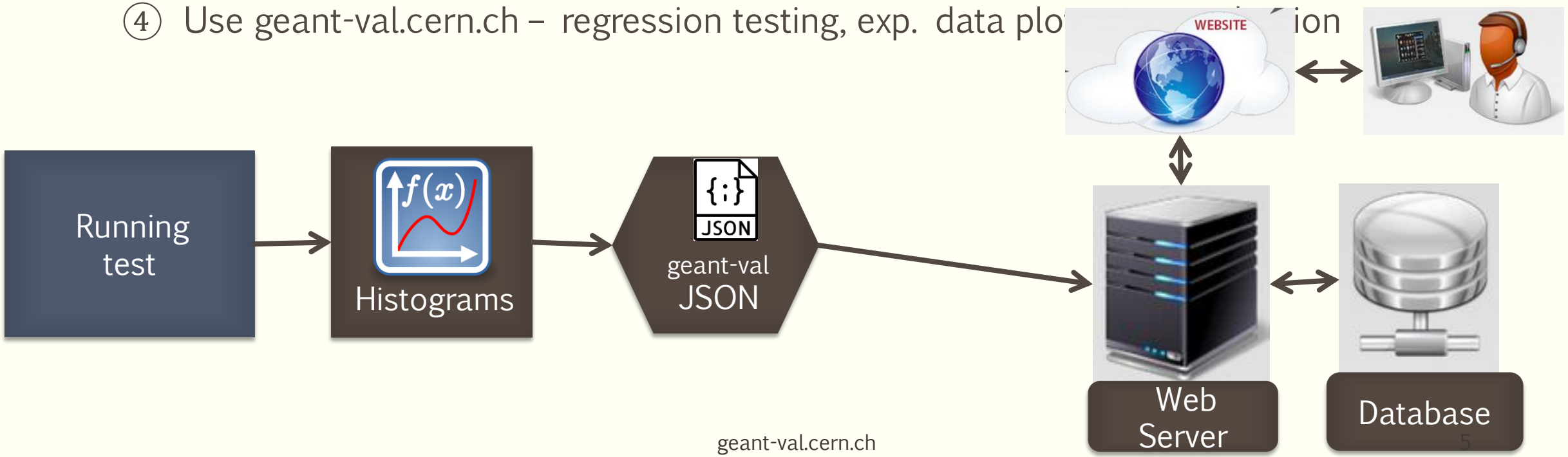
How “geant-val” works

① Run test code locally, using batch systems or with GRID.

② Convert/combine resulted histograms (acsii, ROOT) into “geant-val” JSON objects.

③ Upload “geant-val” JSONs using dedicated python script into “geant-val” (admin)

④ Use geant-val.cern.ch – regression testing, exp. data plo ion



Problems related to compiling and running of tests

There are many tests written and maintained by **different** developers:

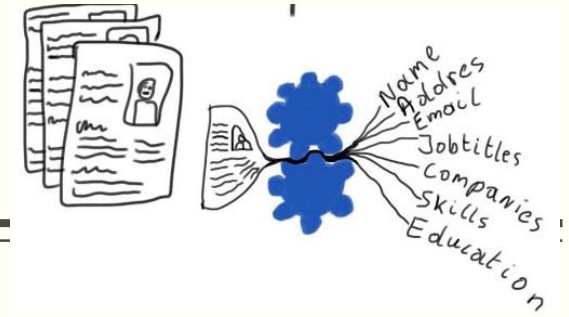
- **different** repos
- **different** compilation ways
- **different** ways to configure and run

Many tests are compiled and placed to CVMFS by Gunter (a lot of work, many thanks!)

There is ongoing development aimed to facilitation of batch/grid submission:

```
python submit.py hadr00 10.2.p02 x86_64-slc6-gcc49-opt
python submit.py sc 10.2.p02 x86_64-slc6-gcc49-opt
```

Parsing of Geant4 test results.



Many Geant4 tests produce results in different formats, without full or proper meta information (observable, axis names) and therefore we see the following solutions:

A) **modification of test source codes** for consistent output format

B) writing of **individual parsers**

At the moment we selected second way - **not difficult, but not that nice and rather annoying**.

```
if 'NA61' in h.GetName():
   yaxis = "D2(SIG)/DP/DTHETA (mb/rad/GeV)"
   testname = "test22-NA61"
```

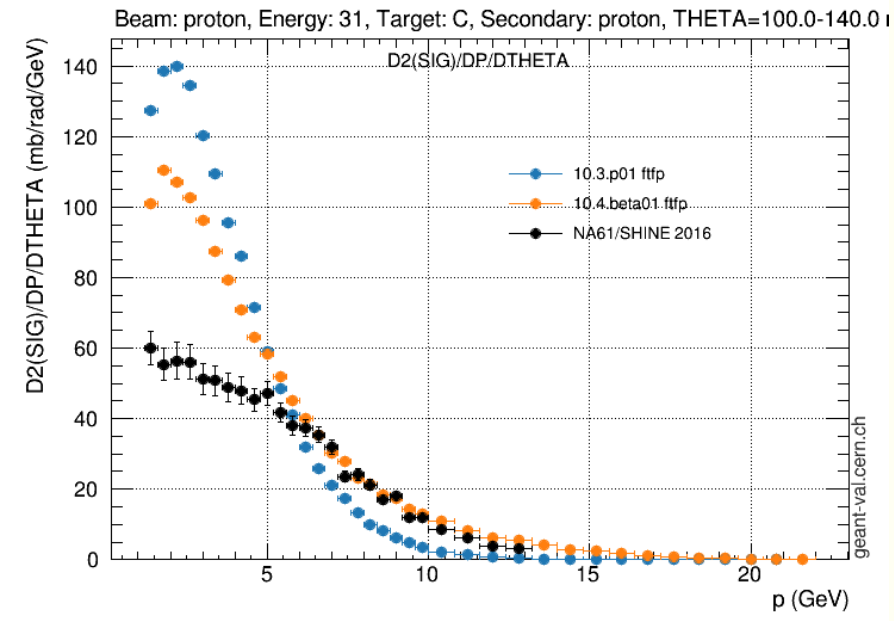
So we have the following parsers:

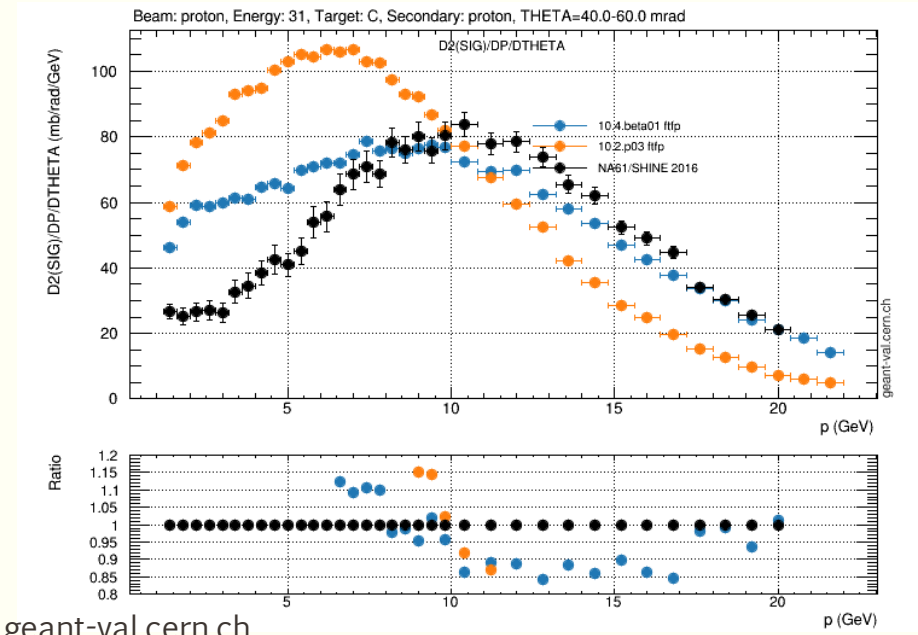
- “simplified calo”: python script to fit and combine results
- “test22”: original test is modified to produce results in ROOT format, dedicated parser of ROOT
- “test46”, “TestEm3”, “hadr00” python ROOT parsers

What “geant-val” can do

- plot MC and experimental data
- create overlaying and ratio plots
  - Release with release
  - Release with experimental data
- produce chi2 table for a given test and primary particle

Observable	Beam	Model	Target	Secondary	Beam energy	Parameters	GEANT4: 10.4.beta01 with GEANT4: 10.2.p03
D2(SIG)/DP/DTHETA	proton	ftfp	C	pi-	31	THETA: 10.0-20.0 mrad	63.9034 (183654 and 177183)
D2(SIG)/DP/DTHETA	proton	ftfp	C	pi-	31	THETA: 0.0-10.0 mrad	66.6596 (183646 and 177175)
D2(SIG)/DP/DTHETA	proton	ftfp	C	proton	31	THETA: 0.0-10.0 mrad	98.2294 (183690 and 177219)
D2(SIG)/DP/DTHETA	proton	ftfp	C	pi+	31	THETA: 0.0-10.0 mrad	147.028 (183647 and 177176)
D2(SIG)/DP/DTHETA	proton	ftfp	C	pi+	31	THETA: 10.0-20.0 mrad	182.231 (183655 and 177184)
D2(SIG)/DP/DTHETA	proton	ftfp	C	pi-	31	THETA: 20.0-40.0 mrad	214.199 (183666 and 177195)
D2(SIG)/DP/DTHETA	proton	ftfp	C	proton	31	THETA: 10.0-20.0 mrad	242.946 (183694 and 177223)
D2(SIG)/DP/DTHETA	proton	ftfp	C	pi-	31	THETA: 40.0-60.0 mrad	370.75 (183682 and 177211)
D2(SIG)/DP/DTHETA	proton	qgsp	C	pi-	31	THETA: 10.0-20.0 mrad	546.643 (183656 and 177185)
D2(SIG)/DP/DTHETA	proton	ftfp	C	proton	31	THETA: 20.0-40.0 mrad	633.366 (183700 and 177229)





Input JSON format

- Simple representation of histogram and corresponding metadata describing configuration/conditions.
- Human readable
- Please note metadata are kept in DB validation tables to implement data integrity.

TH1


```
article:
  inspireId:            593382
mctool:
  name:                 "GEANT4"
  version:              "10.3.ref08"
  model:                "ftfp"
testName:               "test22-NA61"
metadata:
  observableName:       "D2(SIG)/DP/DTHETA"
  reaction:             "particle production"
  targetName:           "C"
  beamParticle:         "pi+"
  beamEnergies:
    0:                  31
  beam_energy_str:      "31"
  secondaryParticle:    "pi+"
  parameters:
    0:
      names:            "THETA"
      values:           "60.0-100.0 mrad"
plotType:               "TH1"
histogram:
  nBins:
    0:                  46
  binEdgeLow:           [46]
  binEdgeHigh:          [46]
  binContent:           [46]
  yStatErrorsPlus:      [46]
  yStatErrorsMinus:     [46]
  ySysErrorsPlus:
  ySysErrorsMinus:
  title:                "D2(SIG)/DP/DOMEGA, pi+ + C -> pi+ + X"
  xAxisName:            "p (GeV)"
  yAxisName:            "D2(SIG)/DP/DTHETA (mb/rad/GeV)"
```


TGraphErrors


```
article:
  inspireId:            -1
mctool:
  name:                 "GEANT4"
  version:              "10.3"
  model:                "FTFP_BERT"
testName:               "simplified calorimeter"
metadata:
  observableName:       "energy response"
  reaction:             "particle production"
  targetName:           "AtlasFCAL"
  beamParticle:         "pi-"
  beamEnergies:         [25]
  beam_energy_str:      "MULTIPLE"
  secondaryParticle:    "None"
  parameters:
plotType:               "SCATTER2D"
chart:
  nPoints:              25
  xValues:              [25]
  yValues:              [25]
  xStatErrorsPlus:      [25]
  yStatErrorsPlus:      [25]
  xStatErrorsMinus:     [25]
  yStatErrorsMinus:     [25]
  xSysErrorsPlus:       [25]
  ySysErrorsPlus:       [25]
  xSysErrorsMinus:      [25]
  ySysErrorsMinus:      [25]
  title:                "energy response"
  xAxisName:            "E_{kin}^{beam}, GeV"
  yAxisName:            "<E_{vis}>/E_{beam}"
```

Access control / CERN SSO authentication

a) for JSON upload with WebAPI

- Only users belonging to geant-val admin e-group

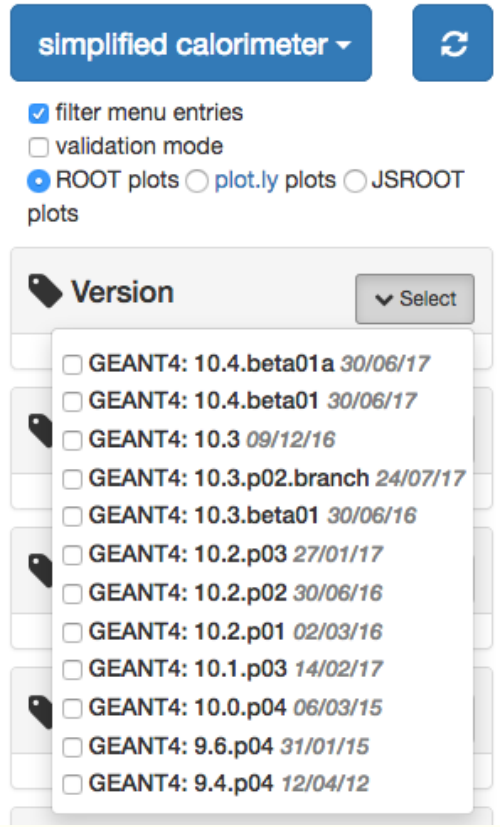
```
python geant_upload.py --krb -j <json file>
```

Alternatively, one can use GRID certificate

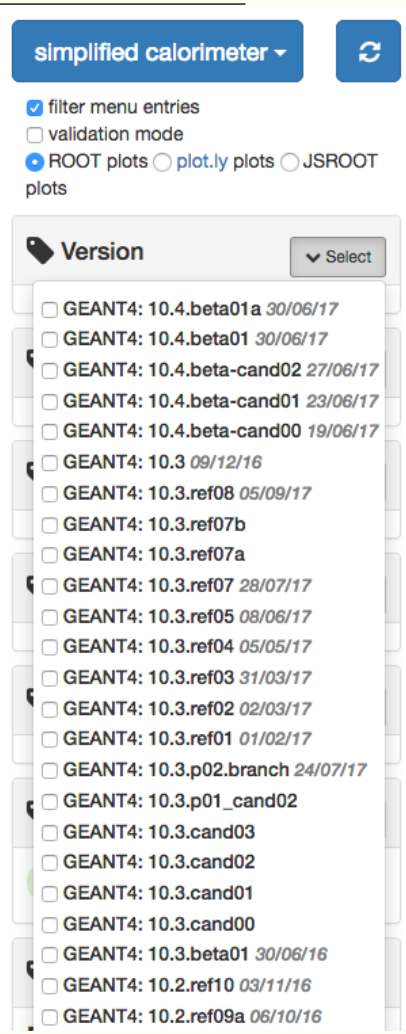
```
python geant_upload.py --key <key file> --cert <cert file> -j <json file>
```

b) restrict access to certain information:

- reference releases
- unvalidated tests



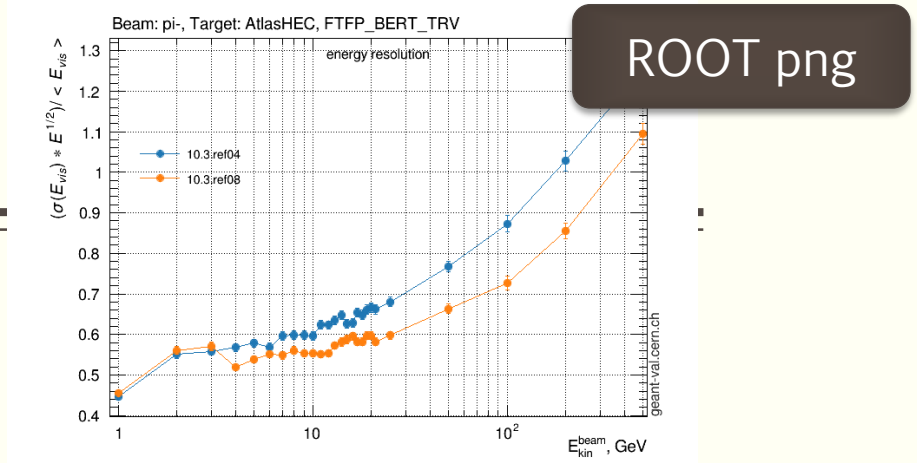
Non authorized view

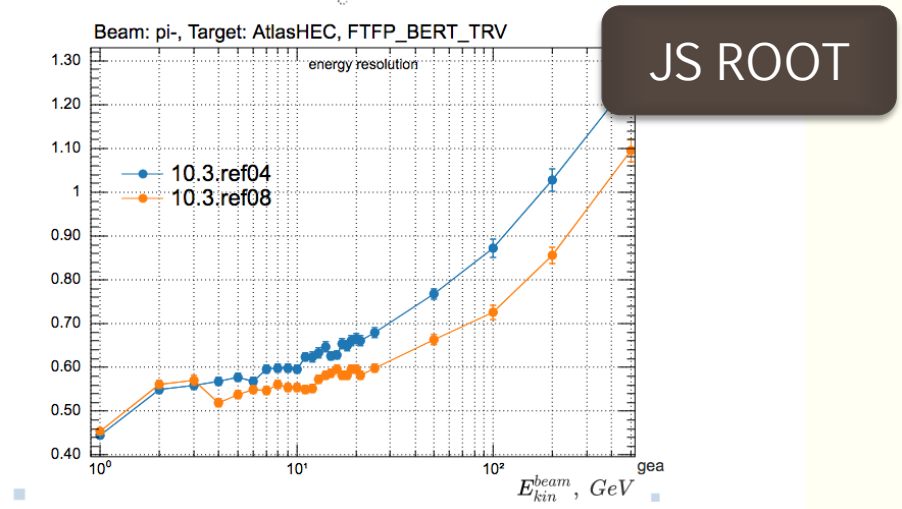


Authorized view

Graphics/Plots

ROOT plots JSROOT plots





Since last Geant4 collaboration meeting we:

- added static ROOT plots
  - png image created **"on the fly" by c++ code** using ROOT 6
  - small size, quick plotting, cached.
- added JavaScript ROOT plots (JSROOT)
  - JS object created "on the fly" by c++ code using JSROOT
  - bigger object but interactive
- "too slow" and "bulky" plot.ly plots deprecated
  - nice plots but "too large" to have more than one
- moved from "default" ROOT colors to "soft" palette taken from "plot.ly"
- Latex formula renderer moved from MathJax to KaTeX:
  - Much much faster! (around 25 times faster)
  - KaTeX only provides a limited subset of the functionality provided by MathJax, but for our purposes it is enough

First step towards statistical evaluation

test22-HARP produces ~700 plots per Geant4 release

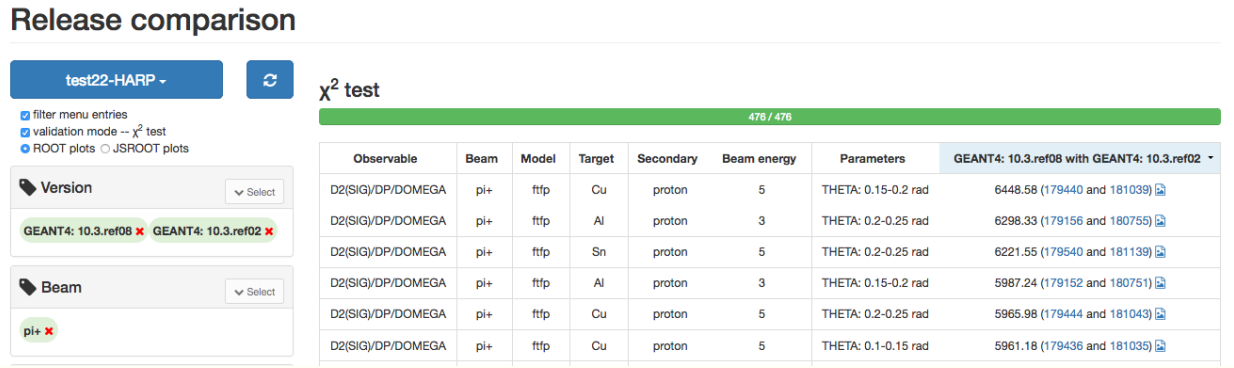
How to compare them with previous release of with experimental data?

How to organize plots for reliable validation?

Introduction of statistical analysis is important!

Possibility to generate table for chi2/ndf between two Geant versions added.

ordering by chi2 value to see histograms with more prominent difference



Release comparison

test22-HARP

filter menu entries
validation mode -- χ^2 test
ROOT plots JSROOT plots

Version Select

GEANT4: 10.3.ref08 × GEANT4: 10.3.ref02 ×

Beam Select

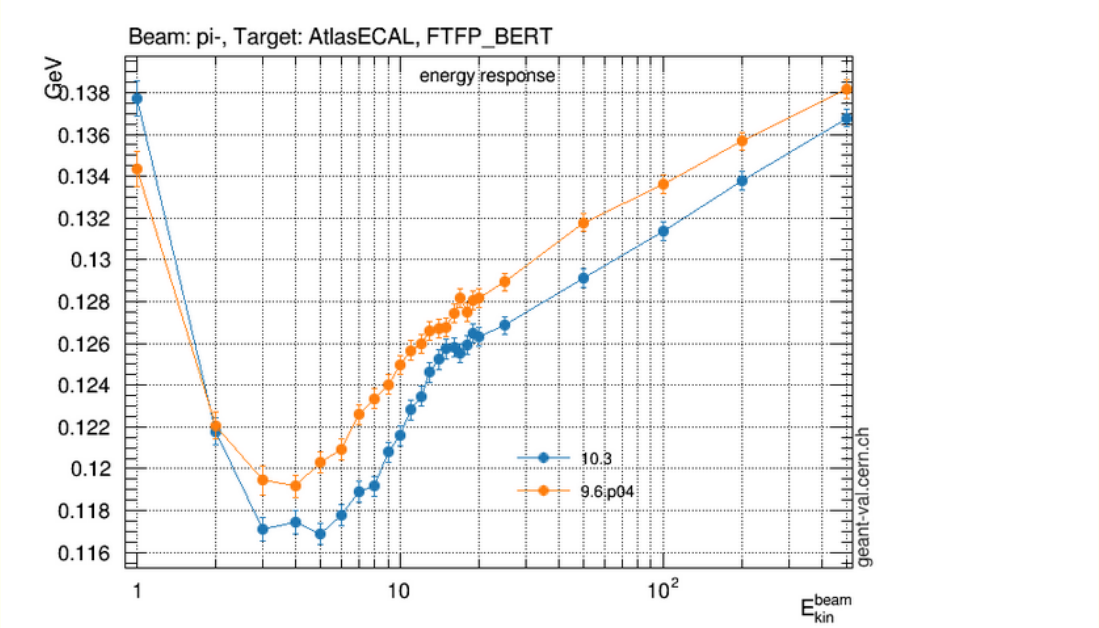
pi+ ×

χ^2 test

476 / 476

Observable	Beam	Model	Target	Secondary	Beam energy	Parameters	GEANT4: 10.3.ref08 with GEANT4: 10.3.ref02
D2(SIG)/DP/DOMEGA	pi+	ftfp	Cu	proton	5	THETA: 0.15-0.2 rad	6448.58 (179440 and 181039)
D2(SIG)/DP/DOMEGA	pi+	ftfp	Al	proton	3	THETA: 0.2-0.25 rad	6298.33 (179156 and 180755)
D2(SIG)/DP/DOMEGA	pi+	ftfp	Sn	proton	5	THETA: 0.2-0.25 rad	6221.55 (179540 and 181139)
D2(SIG)/DP/DOMEGA	pi+	ftfp	Al	proton	3	THETA: 0.15-0.2 rad	5987.24 (179152 and 180751)
D2(SIG)/DP/DOMEGA	pi+	ftfp	Cu	proton	5	THETA: 0.2-0.25 rad	5965.98 (179444 and 181043)
D2(SIG)/DP/DOMEGA	pi+	ftfp	Cu	proton	5	THETA: 0.1-0.15 rad	5961.18 (179436 and 181035)

data download formats



Download list

JSON Files

[GEANT4 10.3 FTFP_BERT, JSON]
[GEANT4 9.6.p04 FTFP_BERT, JSON]

Text Files

[GEANT4 10.3 FTFP_BERT, PLAIN TEXT]
[GEANT4 9.6.p04 FTFP_BERT, PLAIN TEXT]

Plain output format

```
# ID:          169563
# Test:        hadr00
# Tool:        GEANT4 10.3.ref07a
# Beam:        pi+
# Beam Energy: Multiple
# Observable:  elastic cross section
# Secondary:   None
# Target:      Cu
# Parameters:

XVALUE  YVALUE XSTATERRORMINUS XSTATERRORPLUS YSTATERRORMINUS YSTATERRORPLUS XSYSERRORMINUS XSYSERRORPLUS YSYSERRORMINUS YSYSERRORPLUS
0.1012  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1035  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1059  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1084  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1109  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1135  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1161  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1189  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1216  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1245  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1274  0.125  0               0             0.00125         0.00125         0              0             0              0
0.1303  0.125  0               0             0.00125         0.00125         0              0             0              0
```

JSON format


```
JSON  Raw Data  Headers
Copy
  :                169563
  article:
    inspireId:     593382
  mctool:
    name:          "GEANT4"
    version:       "10.3.ref07a"
    model:         "FTFP_BERT"
  testName:        "hadr00"
  metadata:
    observableName: "elastic cross section"
    reaction:      "particle production"
    targetName:    "Cu"
    beamParticle:  "pi+"
    beamEnergies:
      0:           0
      1:           10000000
    beam_energy_str: "MULTIPLE"
    secondaryParticle: "None"
    parameters:
  plotType:        "SCATTER2D"
  chart:
    nPoints:       800
    xValues:
      0:           0.1012
      1:           0.1035
      2:           0.1059
      3:           0.1084
      4:           0.1109
      5:           0.1135
      6:           0.1161
```


- JSON output = JSON input format
- Plain output is not final and will be adapted according to user's requirements

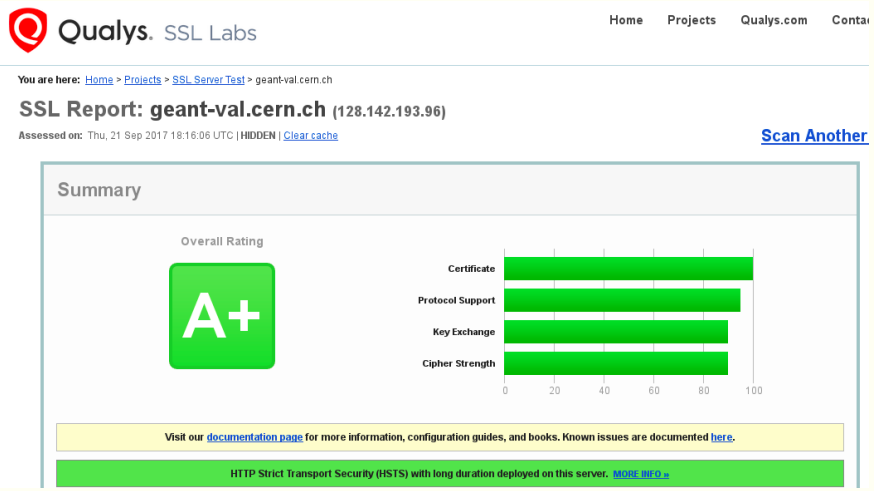
Security

Host security:

- code builds and is encapsulated in docker image
- server process runs as unprivileged user inside docker container
- SSH access to host domain is allowed only by CERN Kerberos ticket

Web security:

- enforced https protocol
- SSL certificates re-issued every 90 days
- A+ (maximal) SSL security level by Qualys SSL scanner



Database security:

- All web APIs scanned for possible SQL injection holes
- Uploading data is allowed only with authorized Kerberos ticket (no other ways to modify data in database)
- Daily database back-up provided by CERN dbondemand service.

Conclusion

We have developed a tool facilitating validation of Geant4 which:

- have **intuitive user interface**
- have **nice graphics**
- Is **secure** (never ending process)

which provides:

- **consistent storage of test results**
- **production of overlaying plots and ratio plots for regression testing**
- **possibility for comparison with experimental data**
- **simple statistical evaluation for regression testing**

short-term plans:

- **continue integration of new tests**
- **continue our work towards more sophisticated statistical analysis**

Backup

- Backup slides

Validation table

Lookup table

The tables below shows available values in database. If you need to add additional one please send email to geant-val[at]cern.ch.

- model names
- version names
- mctool names
- observable names
- particle names
- plot type names
- reaction names
- target names
- test names

> AtlasECAL
> AtlasFCAL
> AtlasHEC
> LhcbECAL
> TileCal
> ECAL+HCAL
> CmsECAL
> hecatlas
> W
> atlasbar

Auxiliary page for G4 exceptions

Simplified calorimeter exceptions

simplified calorimeter

Version

Select From Existing Ones

GEANT4: 10.2.ref02 ✖

GEANT4: 10.2.ref03 ✖

GEANT4: 10.2.ref04 ✖

GEANT4: 10.2.ref04a ✖

Version	had006	test_st...	had012:...	had012:...	HAD_KIN...
GEANT4: 10.2.ref02	1	2	11	185	593
GEANT4: 10.2.ref03	0	2	12	55	630
GEANT4: 10.2.ref04	0	0	5	64	653
GEANT4: 10.2.ref04a	0	0	2	65	653

10 25 50 100

GEANT4_ 10.2.ref02_HAD_KINTRACK_001.txt

```
*** G4Exception : HAD_KINTRACK_001
issued by : G4KineticTrack::Decay
Error condition encountered: phase-space decay failed.
the decaying particle is: f0(1370)
the channel index is: 1 of 6channels
4 daughter particles: pi+ pi- pi0 pi0
*** This is just a warning message. ***

*** G4Exception : HAD_KINTRACK_001
issued by : G4KineticTrack::Decay
Error condition encountered: phase-space decay failed.
the decaying particle is: f0(1370)
the channel index is: 1 of 6channels
4 daughter particles: pi+ pi- pi0 pi0
*** This is just a warning message. ***

*** G4Exception : HAD_KINTRACK_001
issued by : G4KineticTrack::Decay
Error condition encountered: phase-space decay failed.
the decaying particle is: f0(1370)
the channel index is: 1 of 6channels
4 daughter particles: pi+ pi- pi0 pi0
*** This is just a warning message. ***
```

AtlasHEC	19.67%
TileCal	18.85%
CmsECAL	18.85%
LhcbECAL	18.03%
AtlasFCAL	18.03%
AtlasECAL	6.56%
FTFP_BERT_TRV	100%
500 Gev	4.92%
200 Gev	4.92%
100 Gev	4.1%
50 Gev	4.92%
25 Gev	4.92%
20 Gev	3.28%
19 Gev	4.1%
18 Gev	4.1%
17 Gev	4.1%
16 Gev	4.1%
15 Gev	4.1%
14 Gev	4.1%
13 Gev	4.1%
12 Gev	4.1%
11 Gev	3.28%
10 Gev	4.1%
9 Gev	4.1%
8 Gev	4.1%
7 Gev	4.1%
6 Gev	4.92%

Good news! Once we implemented it – all G4Exceptions are disappeared…… ☺