UI and Python Interface

Koichi Murakami (KEK)

Geant4 Collaboration Meeting 2017

Important fix in UI : BZ1989 (2006)

Symptom : In UI terminal,

```
PreInit> ls
terminate called after throwing an instance of
'std::out_of_range' what(): basic_string::substr
Aborted
```

✓ Exception raised, aborted

Cause:

- out-of-range accessing for string in G4VBasicShell::ModifyPath()
- Fixed in interfaces-V10-03-01

Jupyter / ZeroMQ

Jupyter

- http://jupyter.org
- Former IPython notebook
- much more powerful frontend than plain python CLI.
- Notebook works on web browser
  - ✓ inline interactivity : plots, images, …
  - ✓ save and share session logs
  - ✓ familiar with github, nbviewer
- Other external language (R, Julia, SQL,..) can be run on Jupyter frontend

Jupyter Notebook Sample

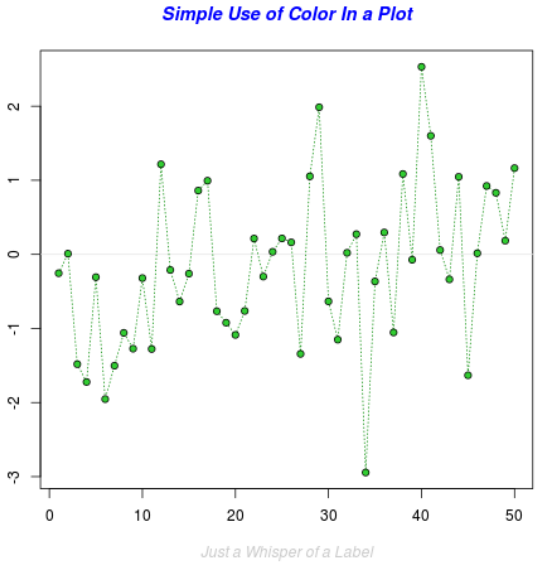
R demo

Here is some code which illustrates some of the differences between R and S graphics capabilities. Note that colors are generally specified by a character string name (taken from the X11 rgb.txt file) and that line textures are given similarly. The parameter "bg" sets the background parameter for the plot and there is also an "fg" parameter which sets the foreground color.

In [1]:
```
require(datasets)

require(grDevices); require(graphics)
```

In [2]:
```
x <- stats::rnorm(50)
opar <- par(bg = "white")
plot(x, ann = FALSE, type = "n") +
abline(h = 0, col = gray(.90)) +
lines(x, col = "green4", lty = "dotted") +
points(x, bg = "limegreen", pch = 21) +
title(main = "Simple Use of Color In a Plot",
      xlab = "Just a Whisper of a Label",
      col.main = "blue", col.lab = gray(.8),
      cex.main = 1.2, cex.lab = 1.0, font.main = 4, font.lab = 3)
```



Out[2]: numeric(0)

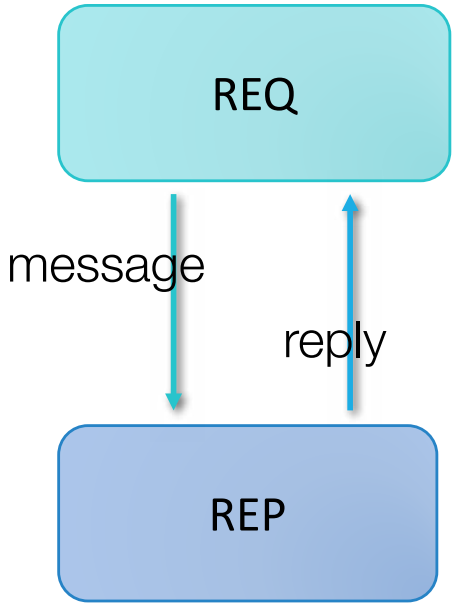
Completion, History, Save

Interactive, Save, Share

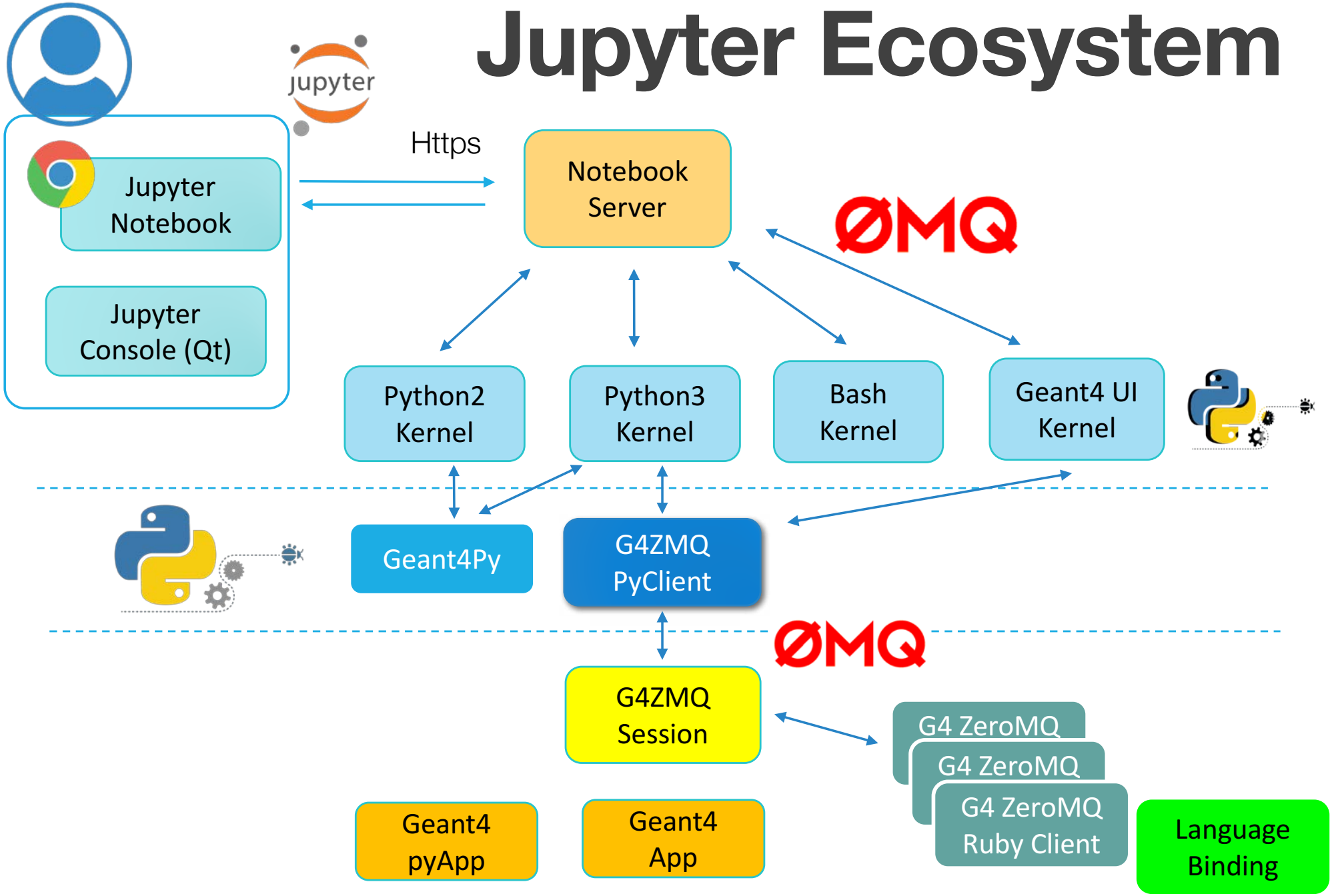
Jupyter Frontend + Language Kernel

ZeroMQ

- light-weight socket API
- support different message patterns
  - ✓ REQ – REP (Request – Reply model)
  - ✓ send a message from client to sever
  - ✓ reply a message from sever to client
- Many language bindings
  - C/C++, Python, Ruby, PHP, Perl, Java, ...



Jupyter Ecosystem



G4ZMQServer

- An alternative UI session like
  - ✓ UI terminal, Qt session, Batch session, ...

```
G4ZMQServer* zmq_session = new G4ZMQServer();
zmq_session-> SetEndpoint(endpoint);
// endpoint is like tcp://127.0.0.1:5555
zmq_session -> SessionStart();
```

- Waiting a message / Receive a message /

  Execute UI (primitive) commands / Send back an output

ZeroMQ client

- Clients can be in any languages that ZeroMQ is bound.
- Simple pyclient module : *g4zmq*

```
>>> import g4zmq
>>> g4zmq.connect()
>>> g4zmq.ls("/run")
>>> g4zmq.help("/run/beamOn")
>>> g4zmq.apply("/run/beamOn 10")
```

- Language bindings :
  - ✓ Python, Julia, Ruby, Perl, PHP, JAVA, ...
  - ✓ http://zeromq.org/bindings:_start

Notes on ZMQ interface

- ZMQ interface is an alternative of UI session.
- You can drive Geant4 app in the same way as UI terminal.
  - ✓ send a UI command, execute, get a response (output)
  - ✓ but more friendly for scripting in different languages
- **CANNOT** directly access to Geant4 objects like **native** Python interface approach (Geant4Py).

IGeant4 :

Geant4 UI Kernel for Jupyter

jupyter console --kernel geant4

You can do:

- ✓ connect to G4 zmq server
- ✓ execute UI / shell commands
- ✓ completion
- ✓ history
- ✓ shell (bash) exec (ex. %shell ls)

jupyter **Untitled** Last Checkpoint: 3 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Geant4

Code

```
In [13]: %connect()

In [14]: ls
Command directory path : /
 Sub-directories :
   /control/   UI control commands.
   /units/   Available units.
   /process/   Process Table control commands.
   /analysis/   ...Title not available...
   /particle/   Particle control commands.
   /geometry/   Geometry control commands.
   /tracking/   TrackingManager and SteppingManager control commands.
   /event/   EventManager control commands.
   /cuts/   Commands for G4VUserPhysicsList.
   /run/   Run control commands.
   /random/   Random number status control commands.
   /material/   Commands for materials
   /physics_lists/   commands related to the physics simulation engine.
   /gun/   Particle Gun control commands.
   /heptst/   Controls for the hadronic energy/momentum test
   /physics_engine/   ...Title not available...
 Commands :

In [15]: /run/beamOn 10

phot:   for  gamma    SubType= 12  BuildTable= 0
      LambdaPrime table from 200 keV to 100 TeV in 61 bins
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
       PhotoElectric :  Emin=        0 eV    Emax=      100 TeV   AngularGenSauterGavrila  FluoActive

compt:   for  gamma    SubType= 13  BuildTable= 1
      Lambda table from 100 eV  to 1 MeV, 7 bins per decade, spline: 1
      LambdaPrime table from 1 MeV to 100 TeV in 56 bins
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
       Klein-Nishina :  Emin=        0 eV    Emax=      100 TeV

conv:   for  gamma    SubType= 14  BuildTable= 1
      Lambda table from 1.022 MeV to 100 TeV, 18 bins per decade, spline: 1
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
        BetheHeitler :  Emin=        0 eV    Emax=       80 GeV
     BetheHeitlerLPM :  Emin=       80 GeV   Emax=      100 TeV

msc:   for e-    SubType= 10

In [11]: /ru
/run/
/run/initialize
/run/beamOn
/run/verbose
/run/printProgress
/run/numberOfThreads
/run/useMaximumLogicalCores
/run/pinAffinity
/run/eventModulo
/run/dumpRegion

In [12]:
```

Jupyter Notebook

Hints for Python Env.

Anaconda with **pyenv** is easiest to build your private Python env.

pyenv: simple Python version management

https://github.com/pyenv/pyenv

Anaconda: Python collection for data analysis

https://docs.anaconda.com

Status - summary

- PoC (Proof of Concept) is GOOD!!
- Not so bad, not so good
- Codes are available in github repos.

  ZMQ interface

  https://github.com/koichi-murakami/zmq-geant4

  Geant4 UI kernel

  https://github.com/koichi-murakami/igeant4

- Revises, testing, examples, documentations are needed.
- Feedback : useful usecases

One more thing

For App config.

How about JSON?

```
{
  "Primary" : {  /* primary setting */
    type : 'gun',     // primary type (gun / beam)
    number : 100,
    // particle starting position (x,y,z) in cm (z >= -100)
    position : [ 5., +.1, +100. ],
   },
...
}
```

Primay/type = gun

Primay/number = 100

Primay/position = [5., 1., 100.]

It is JSON5 and more

How about YAML?