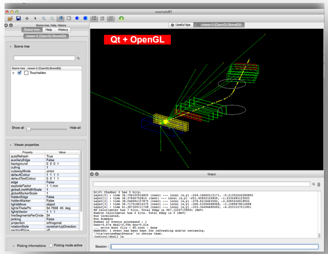
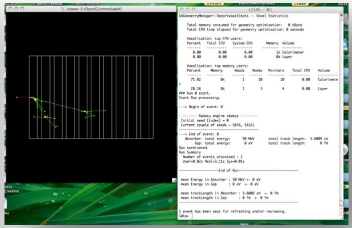
Updates on Geant4Py

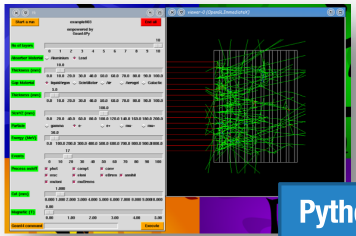
Koichi Murakami (KEK)

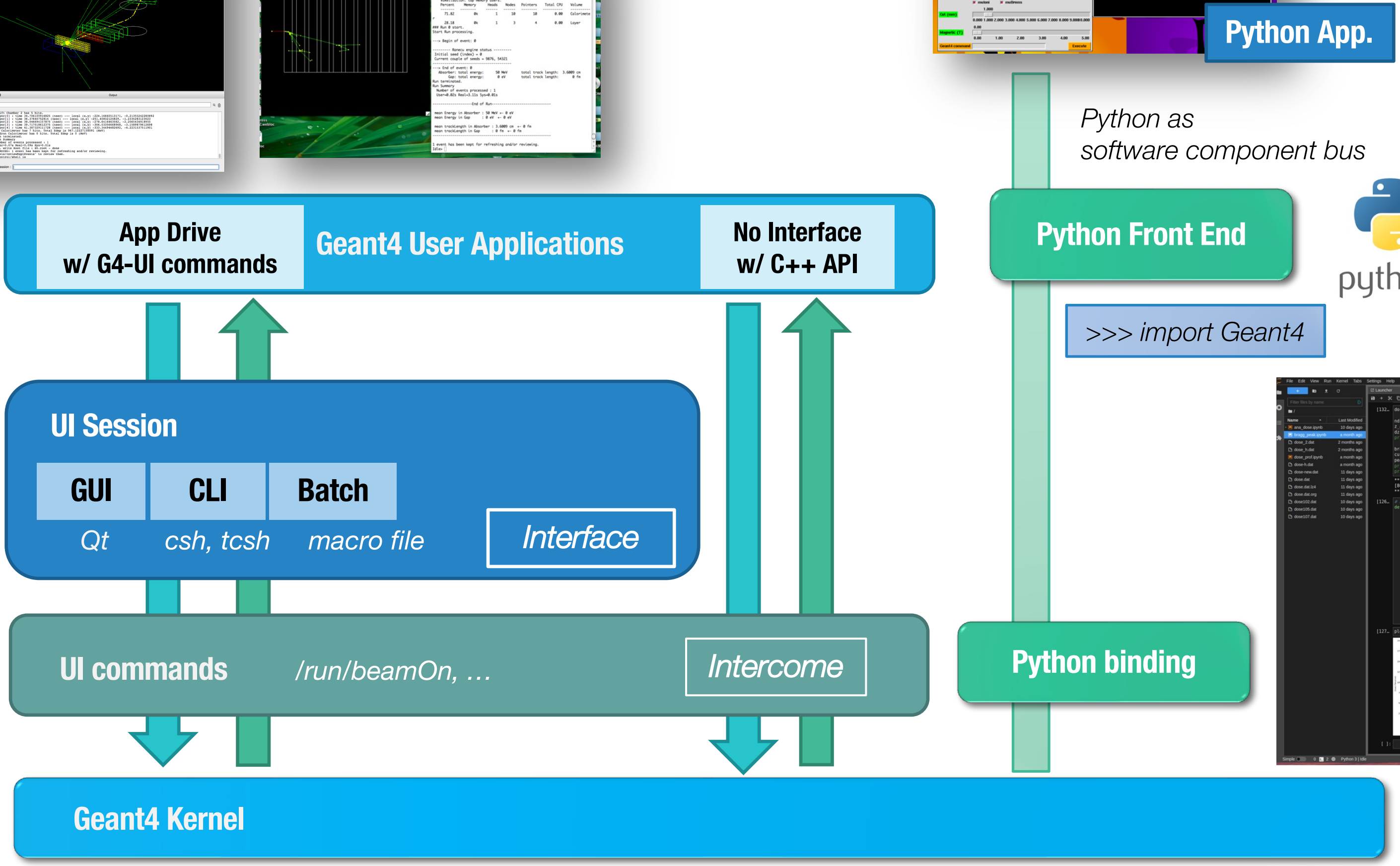
Geant4 Collaboration Meeting 2022

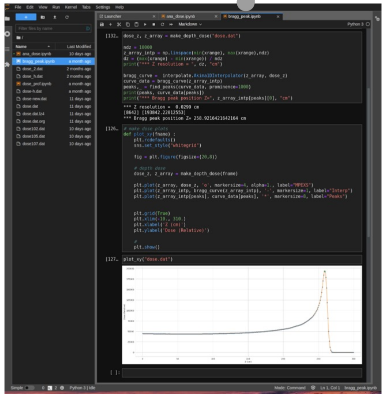
Geant4 UI & App.











Notes on Python Binding

- Python2 : End of life
  - Python2 became End of Life in Apr/2020.
  - Python2 codes will be dropped in the v11 release.
  - Only support Python3 codes

- Boost.python to Pybind11
  - Change C++ binding tool
  - Wrapper approach is very similar to Boost.python (Template base)
  - Header only
  - C++11 (modern C++) support / STL container support

Pybind11

- A binding tool between C++ and Python
- https://github.com/pybind/pybind11
- Header only. Need cmake modules (pybind11)
- Installation:
  - self-install (RH-variants)
  - use apt in Ubuntu
  - use brew in Mac
    - /usr/local (Intel)
    - /opt (Apple Silicon)

What you should know for building

- Install prefix
  - Ex. Geant4Py is installed in $HOME/opt/geant4/geant4py-11.0.0. $HOME/opt/geant4/geant4py-11.0.0/site-packages is needed for PYTHONPATH environment variable as Python module import path.
- Geant4 installation path

  -
    ```
    set(GEANT4_INSTALL $ENV{HOME}/opt/geant4/11.0
                       CACHE STRING "Geant4 installation path")
    ```
- Pybind11 location
  - automatically detected when pybind11 is installed in the system direcrory.

  -
    ```
    set(pybind11_DIR /opt/homebrew/share/cmake/pybind11
                     CACHE STRING "Pybind11 search path")
    ```

Notes on Geant4Py (1)

- There are some tips for running Geant4Py
- LD_PRELOAD (Linux)
  - For TLS memory allocation, we have to preload a Geant4 library.
    - # export LD_PRELOAD=libG4run.so (bash/zsh)
    - # setenv LD_PRELOAD libG4run.so (csh/tcsh)
  - In macOS, this is not necessary.
- Multi-threading feature is off
  - In the current version of Geant4Py, we limit Geant4 in sequential mode forcibly by setting G4FORCE_RUN_MANAGER_TYPE inside __init__.py script.
  - In the next release, we can lift this limit.

Notes on Geant4Py (2)

- Qt5 conflict
  - We recommend building Geant4 without the Qt5 feature to avoid the conflict.
  - If you use the Anaconda version of Ptyhon3, there might be a conflict between the Qt5 libraries. When Geant4Py detects the conflict, it shows the following warning message.

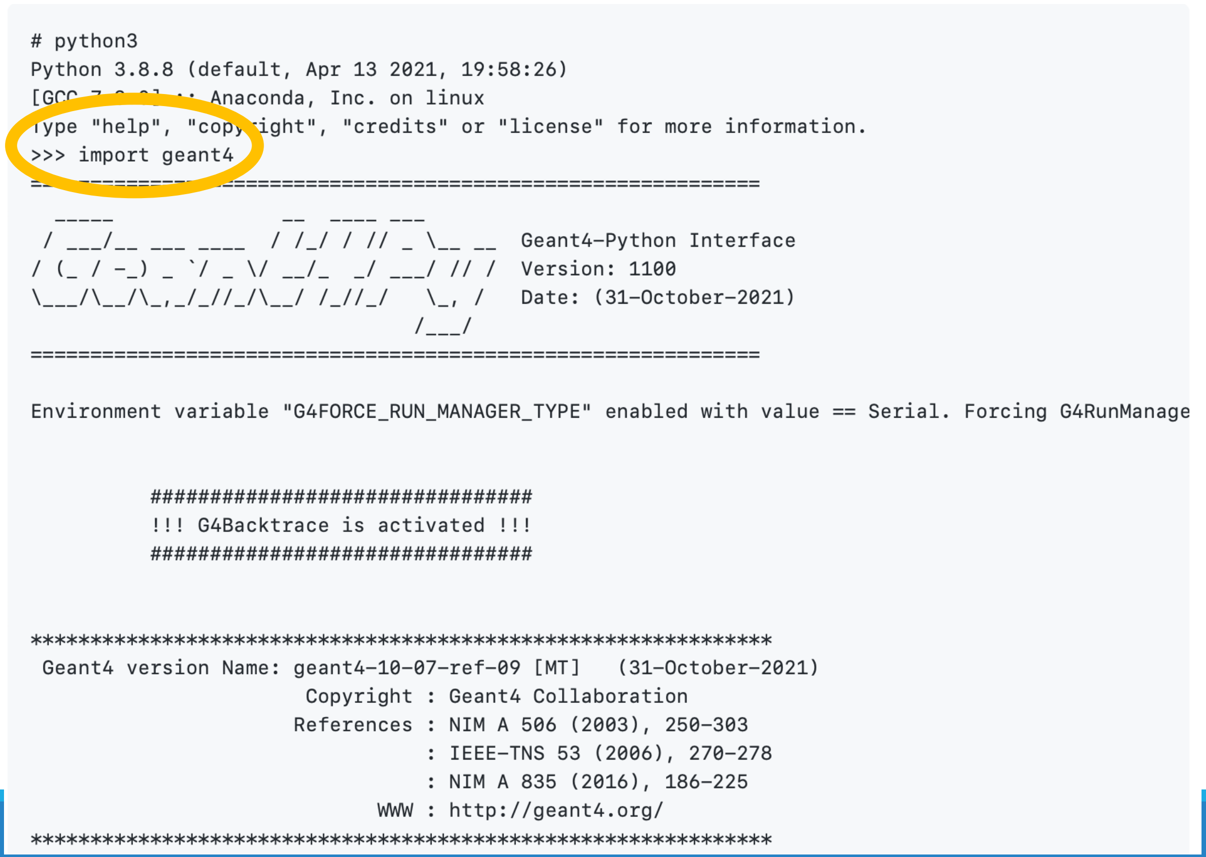
```
#####################################################################
!!! Warning !!!
A non-system python (e.g., Anaconda version of Python) is detected.
If you have a problem with Qt5 library version,
set the environment variables, "G4PY_QT5_PRELOAD = 1"
to preload the system Qt5 library as a temporal solution.
Please consider installing a Geant4 library for Geant4Py
without the Qt feature.
#####################################################################
```

  - Geant4Py will preload the system Qt5 when this environment variable is set.
    - # export G4PY_QT5_PRELOAD=1 (bash/zsh)
    - # setenv G4PY_QT5_PRELOAD 1 (csh/tcsh)
  - Currently, we cannot run Geant4Py on the Anaconda version of Python on Mac. Use the system Python and install the additional packages (Jupyter/numpy/matplotlibt/...) using pip.

Python Runtime

- There are several ways for running Python:
  - System Python3 (not recommended in general)
  - Anaconda Python3 and virtual env versions
  - Use pyenv
  - Fancy frontends:
    - Ipython frontend
    - Jupyter (Jupyter-notebook / Jupyter-lab)
  - +
  - Run as Python scripts

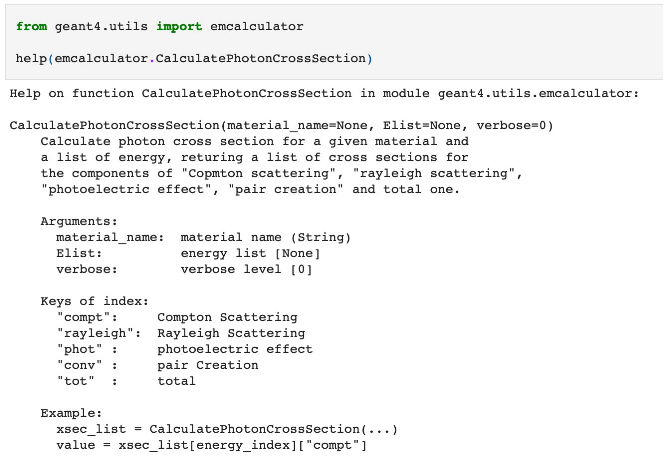
>>> import geant4



```
# python3
Python 3.8.8 (default, Apr 13 2021, 19:58:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import geant4
=============================================================
  _____              __  ____ ___
 / ___/__ ___ ____  / /_/ / // _ \__ __  Geant4-Python Interface
/ (_ / -_) _ `/ _ \/ __/_  _/ ___/ // /  Version: 1100
\___/\__/\_,_/_//_/\__/ /_//_/   \_, /   Date: (31-October-2021)
                                /___/
=============================================================

Environment variable "G4FORCE_RUN_MANAGER_TYPE" enabled with value == Serial. Forcing G4RunManage


          ################################
          !!! G4Backtrace is activated !!!
          ################################


**************************************************************
 Geant4 version Name: geant4-10-07-ref-09 [MT]   (31-October-2021)
                      Copyright : Geant4 Collaboration
                     References : NIM A 506 (2003), 250-303
                                : IEEE-TNS 53 (2006), 270-278
                                : NIM A 835 (2016), 186-225
                            WWW : http://geant4.org/
**************************************************************
```

Utils modules

- In addition to G4 classes, there are some utility user classes:
  - Used for mockup and utilities
  - Geometry:
    - SimpleBox: Mockup usage. Mateirl can be changed.
      - `>>> from geant4.utils import SimpleBox`
    - WaterPhantom: Medical usage. Size and location can be changed.
  - PrimaryGeneratorAction:
    - ParticleGun / GPS / MedicalBeam
  - Physics:
    - emcalculator: CalculatePhtonCrossSection(), CalculateDEDX()



```
from geant4.utils import emcalculator

help(emcalculator.CalculatePhotonCrossSection)
```

```
Help on function CalculatePhotonCrossSection in module geant4.utils.emcalculator:

CalculatePhotonCrossSection(material_name=None, Elist=None, verbose=0)
    Calculate photon cross section for a given material and
    a list of energy, returing a list of cross sections for
    the components of "Copmton scattering", "rayleigh scattering",
    "photoelectric effect", "pair creation" and total one.

    Arguments:
      material_name:  material name (String)
      Elist:          energy list [None]
      verbose:        verbose level [0]

    Keys of index:
      "compt":     Compton Scattering
      "rayleigh":  Rayleigh Scattering
      "phot" :     photoelectric effect
      "conv" :     pair Creation
      "tot"  :     total

    Example:
      xsec_list = CalculatePhotonCrossSection(...)
      value = xsec_list[energy_index]["compt"]
```

Examples with Jupyter (1)

- There are 3 examples with Jupyter .ipynb files:
- exampleB1
  - This example has the same capability as Geant4 basic example B1.
  - The geometry is implemented in C++ and exported to a Python module, which shows how to export your C++ component to Python. (Thin wrapping approach)
- phantom_dose
  - This example shows a practical application. It contains a complete chain of simulation and analysis processes.
  - We calculate dose distributions in a water phantom for electron and proton beams.
  - Voxel doses are scored with the command-line scoring capability and stored into CSV files.
  - This data is analyzed with Pandas and Matplotlib Python tools. Finally, dose maps and depth dose curves are obtained.

Examples with Jupyter (2)

- emplot
  - This example shows how to retrieve the photon cross-sections and stopping powers of charged particles.
  - It prepares a mockup (geom/pl/primary), then changes the target materials.
  - The EM calculator can calculate a cross-section for each process and stopping powers.
  - For stopping power, the ionization and bremsstrahlung components can be calculated for electrons.
  - The example includes plots by Matplotlib.

- Quick view of actual notes…

Tips (1)

- Global variables and functions
  - In Geant4Py, we instantiate Geant4 singleton manager objects (e.g., G4RunManager) at the timing of module loading. These instances are assigned to global variables inside Python, that starts with g character.
    - RunManger : gRunManager
    - EventManager : gEventManager
    - NistManager : gNistManager
    - ScoringManager : gScoringManager
    - …
  - Also, some useful methods are defined as global functions.
    - gStartUISession, StartUISession : Start UI terminal
    - gControlExecute, ControlExecute : Execute a macro file
    - gApplyUICommand, ApplyUICommand : Execute UI command
    - gGetCurrentValue, GetCurrentValue : Get current value of UI command

Tips (2)

- Python Objects
  - Python variables are automatically managed, which means a local variable is automatically deleted on the Python side.
  - This mechanism is different from objects allocated in C++. Some classes are taken care as nodeleted objects in Geant4Py, but still not perfect. If there is a weird behavior (seg. fault), set the Python variable as global.

- Important:
  - An object of user inherited class in Python should be set as global.

Future development

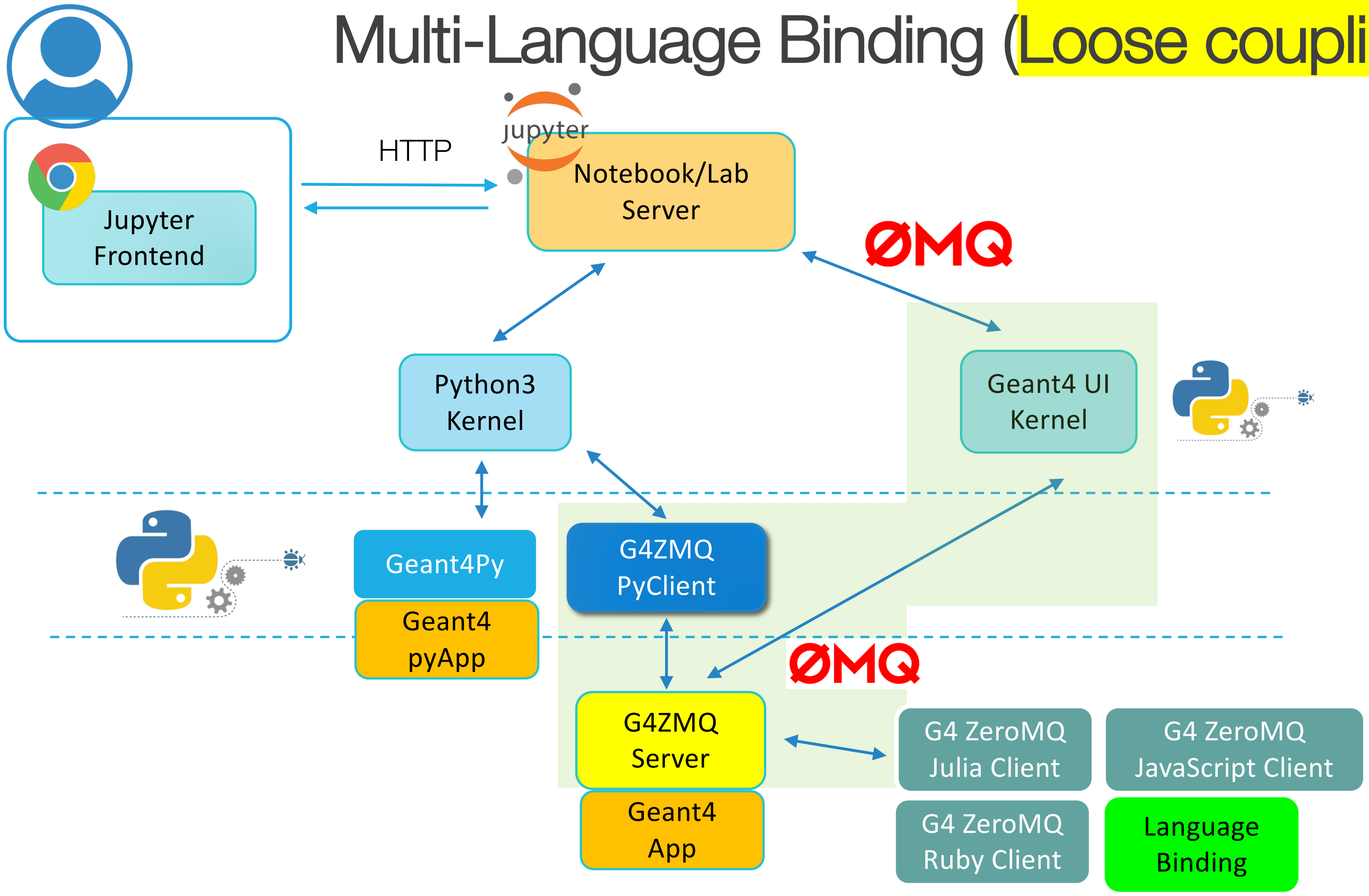
- Multi-thread mode as default
  - Number of threads = 1 still as default

- Introduce lock mechanism for cout
  - provide cout buffer for each thread and selector by thread
  - also implement for UIterminal

- Export more classes:
  - especially for scoring part
  - No aim for full python app : thin wrapping approach

- Find smarter way for tricks
  - LD_PRELOAD
  - Qt5 (Anaconda) conflict

Multi-Language Binding (Loose coupling)



Jupyter Frontend
HTTP
jupyter
Notebook/Lab Server
ØMQ
Python3 Kernel
Geant4 UI Kernel
Geant4Py
Geant4 pyApp
G4ZMQ PyClient
ØMQ
G4ZMQ Server
Geant4 App
G4 ZeroMQ Julia Client
G4 ZeroMQ JavaScript Client
G4 ZeroMQ Ruby Client
Language Binding