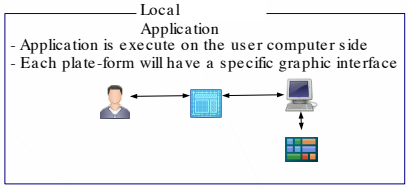
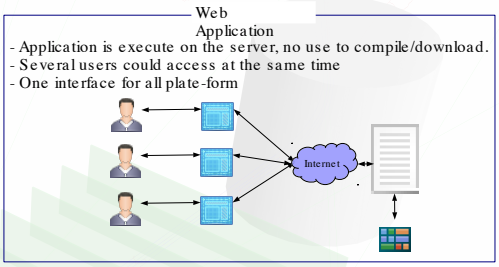
Geant 4 application in a web browser

JDEV - 4 Sept. 2013

The GEANT4 Visualisation System is a **multi-driver** graphics system designed to serve the GEANT4 Simulation Toolkit. It is aimed at the visualisation of GEANT4 data, primarily detector descriptions and simulated particle trajectories and hits. It can handle a variety of graphical technologies simultaneously and interchangeably, allowing the user to choose the visual representation most appropriate to requirements.

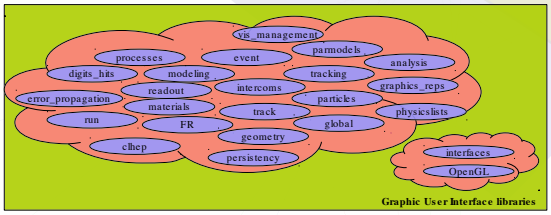
Nowadays, one of theses requirements is to have a Geant4 application embedded in a **Web Browser**





Geant4 library has been design to separate graphical user interface in separated libraries. In order to have good perfomances, we don't want to redesign all Geant4 libraries for the Web, but only the User Interface. For that, we will have to create a new graphical driver library to handle user interactions and communicated with the other Geant4 libraries. For that purpose, we will use the Wt framework.

Developping a web application with **Wt framework** does not differ in structure from an application developed for other graphical user interface systems, such as Windows or X11. Most importantly, the entire application is written in only one compiled language (Geant4 langage is mainly C++), from which the library generates the necessary HTML, Javascript, CGI, and AJAX code. The responsibility of writing browser-portable web applications is transferred to the library implementation. For example, if available, Wt will maximally use JavaScript and AJAX, but applications developed with Wt are also able to function correctly when AJAX is not available, or when JavaScript is disabled, reverting to a completely different technology for communication between browser and server. An application developed with Wt is technology-agnostic.



Geant4 libraries

Thanks to Wt framework, we do not have to write any HTML5, Javascript, or other CSS, you write your interface in C++ language. All User Interface interactions will be done by the framework. Hopefully, the syntax of this frameworks looks like the Qt framework.

```
/** Create the History ToolBox Widget
 */
Wt::WWidget* G4UIWt::CreateHistoryTBWidget(
)
{
  fHistoryTBWidget = new Wt::WPanel();
  fHistoryTBWidget->setCollapsed(true);
  fHistoryTBWidget->setCollapsible(true);

  fHistoryTBTableList = new Wt::WSelectionBox();
  fHistoryTBTableList->setSelectionMode(Wt::SingleSelection);
  fHistoryTBTableList->changed().connect(this,&G4UIWt::CommandHistoryCallback);

  fHistoryTBWidget->setCentralWidget(fHistoryTBTableList);
  return fHistoryTBWidget;
}
```

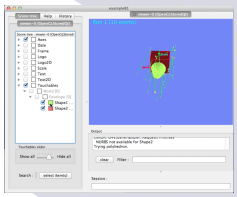
```
/** Create the History ToolBox Widget
 */
WWidget* G4UIQt::CreateHistoryTBWidget(
)
{
  fHistoryTBWidget = new QWidget();

  QVBoxLayout* vlayoutHistoryTB = new QVBoxLayout();
  fHistoryTBTableList = new QListWidget();
  fHistoryTBTableList->setSelectionMode(QAbstractItemView::SingleSelection);
  connect(fHistoryTBTableList, SIGNAL(itemSelectionChanged()), SLOT(CommandHistoryCallback()));
  fHistoryTBTableList->installEventFilter(this);

  layoutHistoryTB->addWidget(fHistoryTBTableList);

  fHistoryTBWidget->setLayout(layoutHistoryTB);
  return fHistoryTBWidget;
}
```

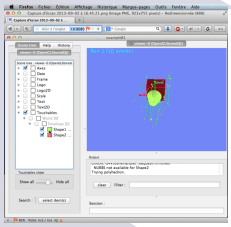
What could it be useful for ?

- Running big simulation on servers
- Teaching physics with Geant4 web application
- Share applications
- Available everywhere, only need a recent navigator

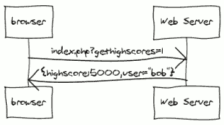
What is the status of this project ?

- Still in development (September 2013)
- User Interface (except OpenGL part) is now integrate and work perfectly, you can interact with Geant4 kenerl thanks to the Web browser :-)
- OpenGL part require a bit more work, Geant4 OpenGL is based on OpenGL 2.1, but on a Web browser, Graphic layer is based on OpenGL ES





C++ program integrated with web server



JDEV2013

Laurent Garnier: garnier@lal.in2p3.fr

IN2P3
Institut National de Physique Nucléaire et de Physique des Particules

LAL
Laboratoire de l'Accélérateur Linéaire