GenApp: The CCP-SAS deployment framework

Emre H Brookes

4th CCP-SAS Project Meeting

Cardiff, Wales, UK
19 June 2017

SCHOOL OF MEDICINE

UT HEALTH
SCIENCE CENTER®

DEPARTMENT OF BIOCHEMISTRY

Outline

- GenApp Background
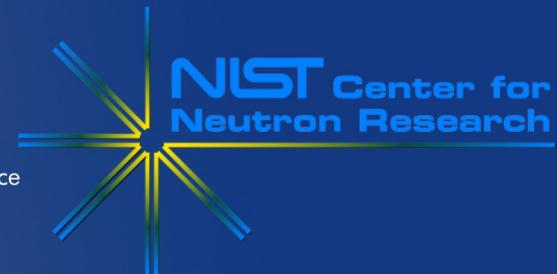
- Current

- Future

Outline

- **GenApp Background**

- Current

- Future

CCP-SAS

- SASSIE http://www.smallangles.net/sassie
  - Joseph Curtis et al.
  - **PYTHON**
  - **includes wrapped binary executables**
- SCT/SCTPL/HYDRO http://www.ucl.ac.uk/smb/perkins
  - Steve Perkins et al.
  - Structural Immunology Group at University College London
  - **FORTRAN**
- US-SOMO http://somo.uthscsa.edu
  - Emre Brookes et al.
  - **C++/Qt**
  - **includes wrapped binary executables**
- attract others …
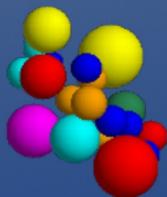
Considerations

- **Ease of deployment** in an ever-evolving software environment landscape

- Legacy and frequently specific lab developed codes

- Labs frequently can not afford a dedicated software team nor the cost in time and funding

Goals
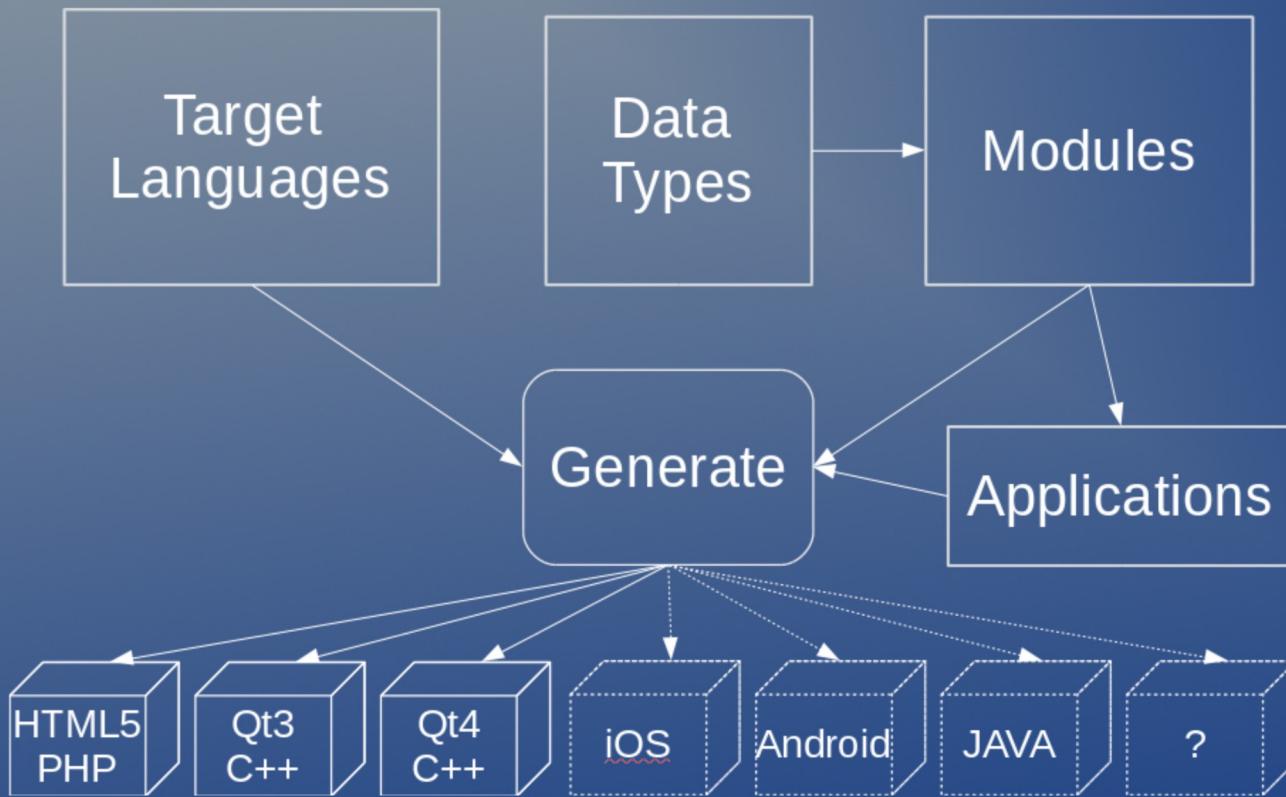
- Simplify application creation by generalizing the process for
    - *Users* of the generated application
    - *Research scientists* with code to deploy
    - *System administrators* managing the applications and their environments
   - *Computer scientists* developing the tool

Simplify

- Simplify application creation by generalizing the process for

    - *Users* of the generated application

    - *Research scientists* with code to deploy

    - *System administrators* managing the applications and their environments

   - *Computer scientists* developing the tool

- Encapsulate & reuse

An Open Extensible Multi-Target Application Generation Tool for Simple Rapid Deployment of Multi-Scale Scientific Codes

Brookes, E. H. XSEDE 14 Atlanta
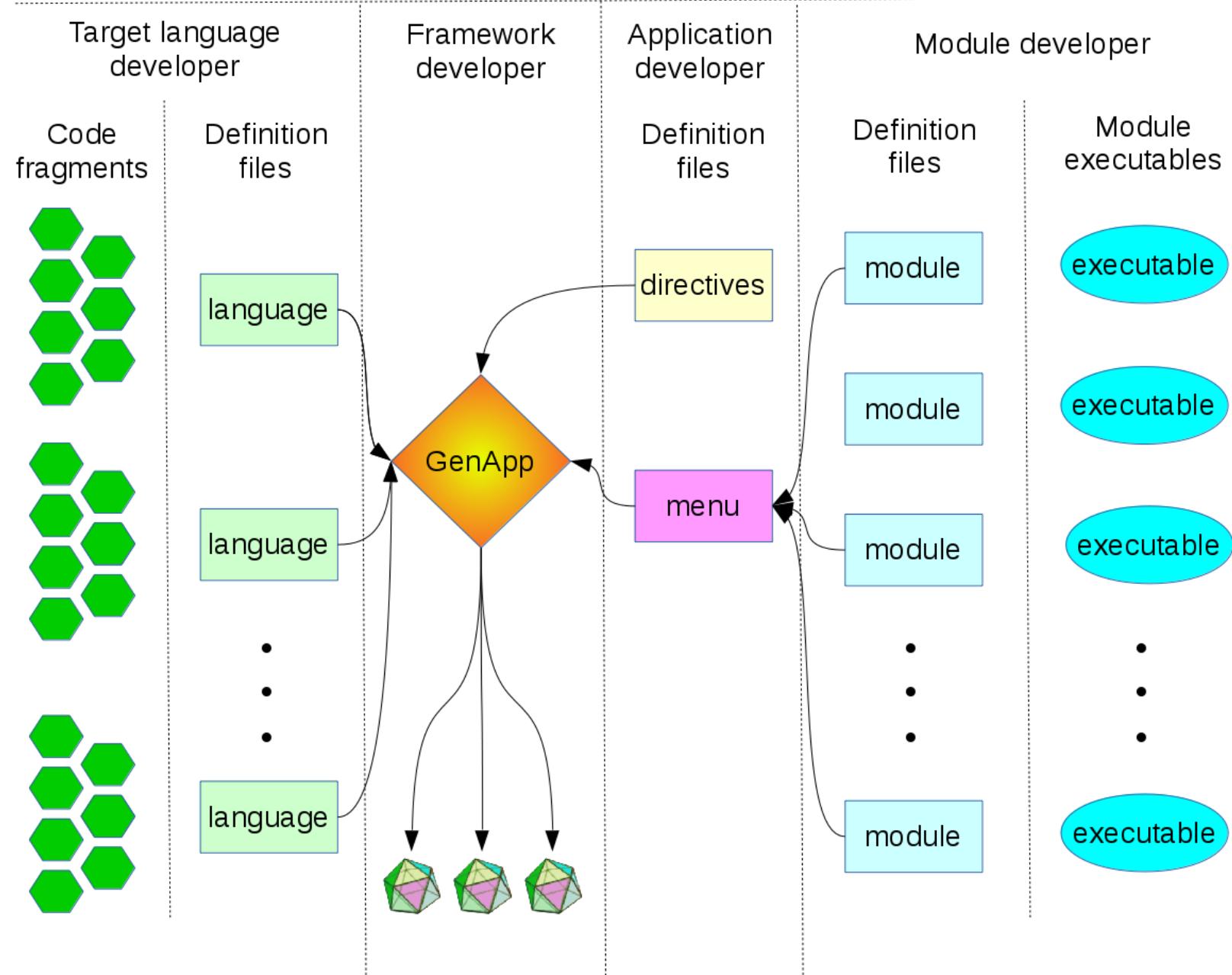
- JSON
- Global directives & application menu and configuration
- Module I/O definitions
- Module messaging
- Language assembly descriptions

Brookes, E H, 2014, An Open Extensible Multi-Target Application Generation Tool for Simple Rapid Deployment of Multi-Scale Scientific Codes, ACM, 10.1145/2616498.2616560

Computer Scientists maintaining the framework

Researchers with code to deploy

Target language developer | Framework developer | Application developer | Module developer

Code fragments | Definition files | Definition files | Definition files | Module executables

language

language

language

directives

GenApp

menu

module — executable

module — executable

module — executable

module — executable

Brookes, E. H. CCP-SAS @ Cardiff – 19 June 2017

GenApp : Simplify

- User of the *Gen*erated *App*lication
  - Consistently generated interface
  - Easy to use to get the "job" done
  - User feedback essential

GenApp : Simplify

- Research scientist with code to deploy

  - Setup your global application definitions

  - Define your module inputs and outputs

  - Wrap your code to read inputs and produce outputs in the defined format

  - Run the tool to *Gen*erate *App*lications

- When new "target languages" are developed, they can be generated on existing modules.

```
def einstein(mass, speed_of_light):
    return mass*(speed_of_light**2.0)
```

energy.py

To "GenApp" you need to write two files:

A text file describing your input / output

modules/energy.json

And

A script to handle input and output between
the "App" and your program

bin/energy

Run a pre-installed script

Done ⟶

**Web-App, QT, Java, Android, iOS,
...**

> genapp.pl

Credit: J. Curtis

Encountering GenApp (... as a researcher ...)

Underlying codes written in different languages, can be GUI or command line applications:

- C++, Python, Fortran etc.
- C++ wrapped in Python (WillItFit); C++ bundled with Qt (US-SOMO)

↓

"Divorce" GUI from the computational component:

- Create a "command-line" analogue: [./program < {arguments}]

↓

<u>Wrap a command-line application:</u>

- Application arguments and output are JSON strings of key-value pairs, {"binsize_id": 0.2, "filename_id": "trajectory.dcd" etc.}
- These key-value pair are described in the application (module) definition file
- Different languages can be used to parse JSON input and transfer it to the underlying application (Python, Perl, C++)

↓

"GenApp", i.e. compile application and generate a web site.

*Credit: A. Savelyev*

GenApp : Simplify
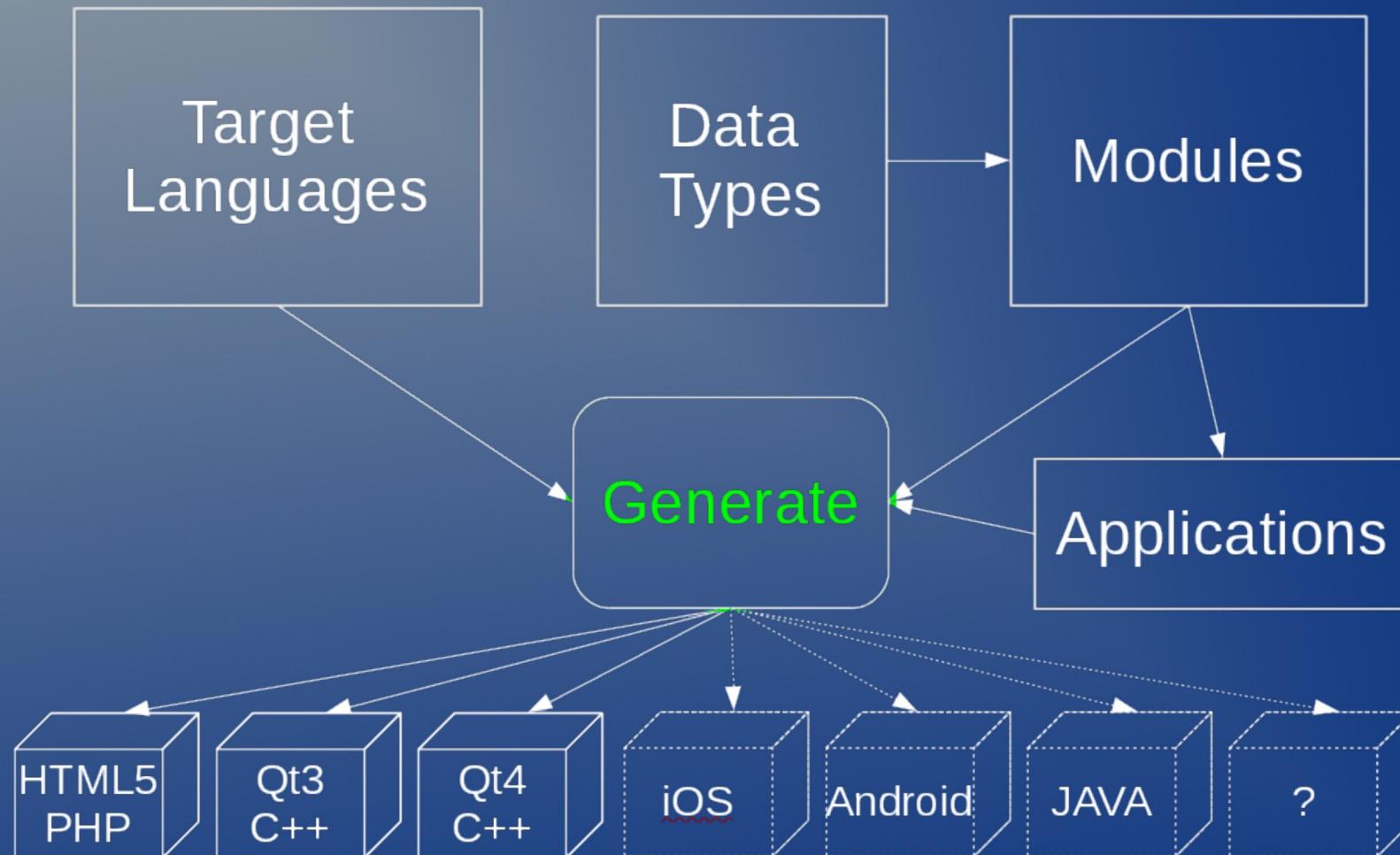
- System administrators
  - Vortex shedding site
    - Primary site ran out disk space
    - Changed host hardware and operating system, running from a fresh install in a couple of hours
  - Setting up a demo VM site
    - New instance was needed with short notice
      - Jetstream host issues with previous
    - Has a fresh VM "machine" installed from "scratch" within a couple of hours
      - Setting up VM, downloading GenApp, running install utility, running demo user creation utility.
    - Hardest part was printing and cutting the user / password notes.

GenApp : Simplify

- Computer scientist developing the tool
  - Easily rapidly extensible
  - Requires CS expertise & understanding of the framework
  - Schema-less
    - Add fields to definition files and use at will
  - GSoC 2014 student A. Kapoor created the new JAVA target language in a week

GenApp : What is it?

- An assembler of fragments of code directed by definition files

# Brief Demo

- quick look at a few websites
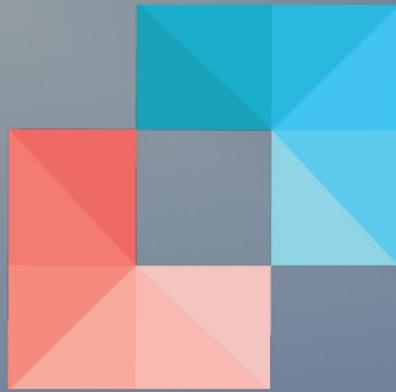
GenApp: Where can jobs actually run?

GenApp: Where can jobs actually run?

- Where the assembled target language fragments allow

GenApp: Where can jobs actually run?
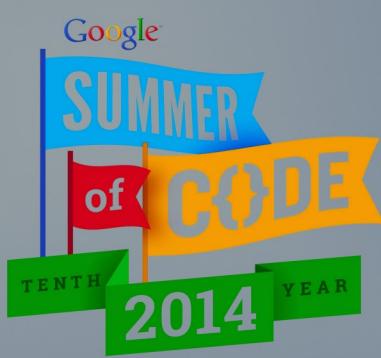
- Where the assembled target language fragments allow
    - qt3/4/5 java
        - locally
        - remotely via Airavata
    - html5/php & android
        - on the server host
        - on an ssh server accessible resource
        - remotely via Airavata
        - elastically via OpenStack integration

APACHE AIRAVATA

- Apache Airavata™ is a software framework that enables you to compose, manage, execute, and monitor large scale applications and workflows on distributed computing resources such as local clusters, supercomputers,computational grids, and computing clouds.

http://airavata.apache.org
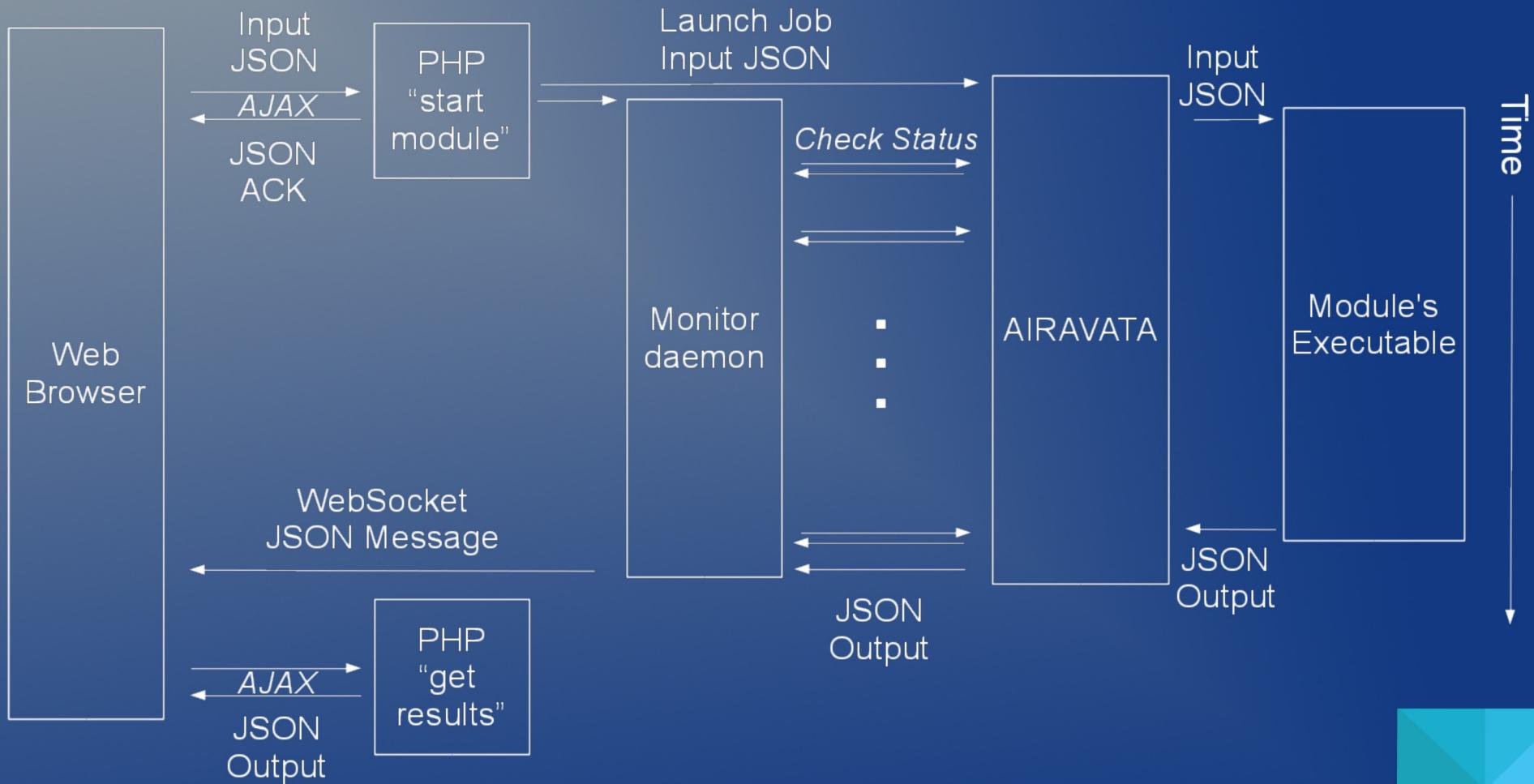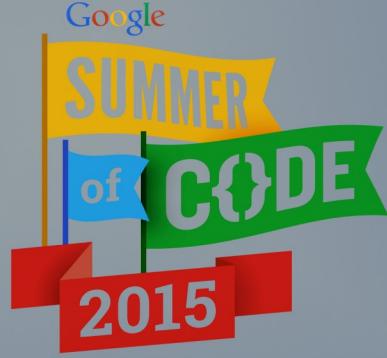
GSoC 2014
GenApp Enhanced execution model HTML5/PHP with Apache Airavata
Nadeem Anjum / Mentor: Suresh Marru

Brookes, E H, Anjum, N, Curtis, J E, Marru, S, Singh, R, Pierce, M, 2014, GenApp module execution and airavata integration, IEEE Press, 10.1109/GCE.2014.12

GSoC 2015
Further Airavata integration

- Abishek Kapoor

    - Integrated target language HTML5 with Airavata PHP Thrift interface

    - Created new target language JAVA and integrated with Airavata JAVA Thrift interface

        - i.e. created a "language".json : java.json and all the code fragments required to produce working java code
        - a "Framework developer" role

- Pryanshu Patra

    - Integrated target language Qt4, Qt5 with Airavata C++ Thrift interface

    - Created target language Qt5/Android with Airavata C++ Thrift interface

Brookes, E. H. CCP-SAS @ Cardiff – 19 June 2017
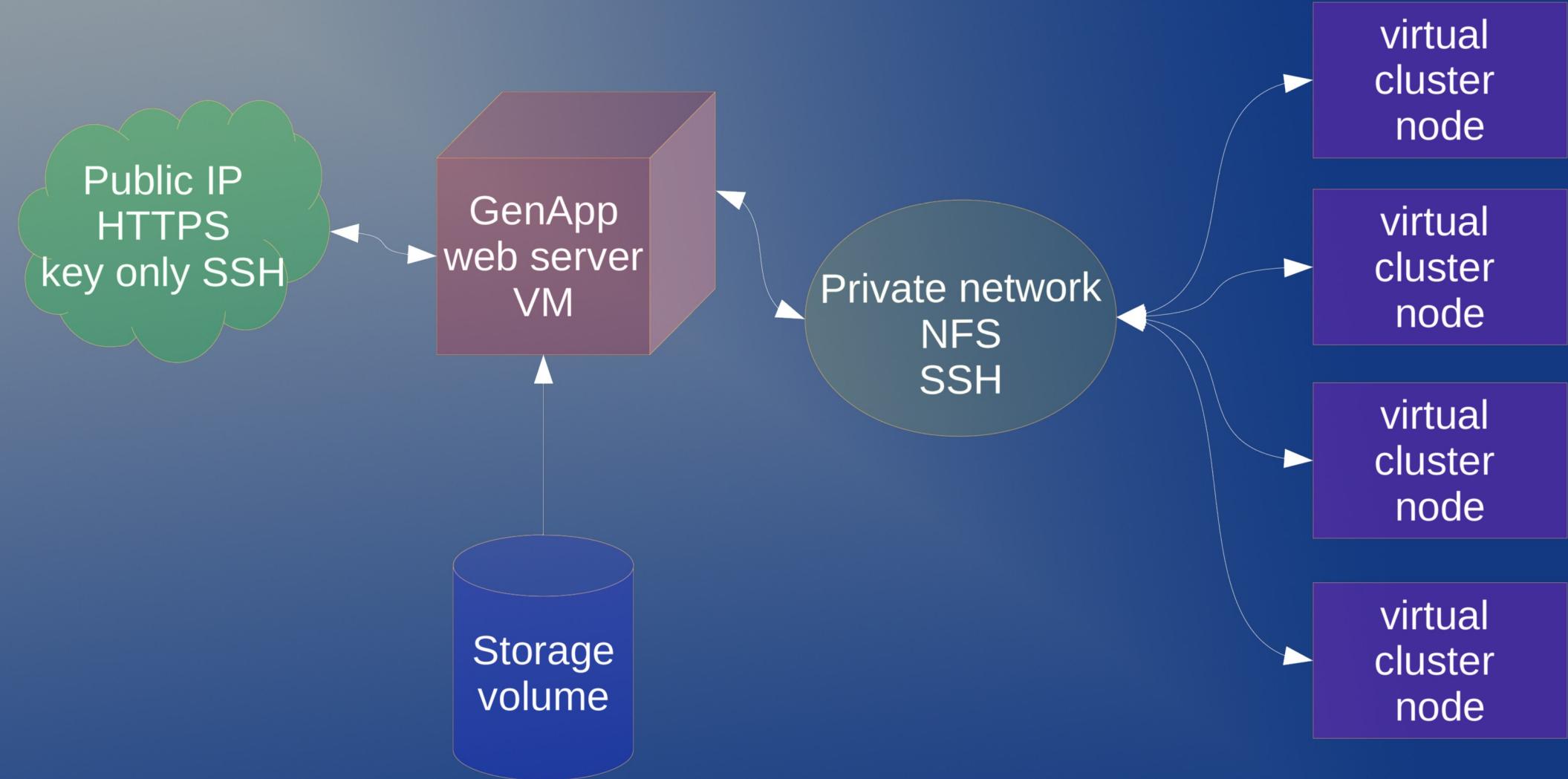
appconfig.json with Airavata resources

```json
{
    ,"resources" : {
        "local"     : "",
        ,"compute0" : "ssh compute-0-0"
        ,"compute1" : "ssh compute-0-1"
        ,"airavata" : {
            "run" : "airavatarun",
            "properties"    : {
                "server"                : "gwXX.iu.xsede.org",
                "port"                  : 8930,
                "timeout"               : 5000,
                "credentialStoreToken"  : "XXXXXX-XXXX-XXXX-XXXXXXX",
                "gateway"               : "genAppGateway",
                "gatewayName"           : "some_gateway_name",
                "login"                 : "admin",
                "projectAccount"        : "projectaccountname",
                "email"                 : "someemail@somewhere"
            },
            "resources"    : [
                {
                    "host"            : "localhost",
                    "description"     : "Localhost",
                    "executable_path" : "/home4/xyz/bin/",
                }
                ,{
                    "host"            : "stampede.tacc.utexas.edu",
                    "description"     : "Stampede",
                    "executable_path" : "/users/abc/bin/"
                }
            ]
        }
    }
    ,"resourcedefault"    : "local"
    ,"submitpolicy"       : "login"
```
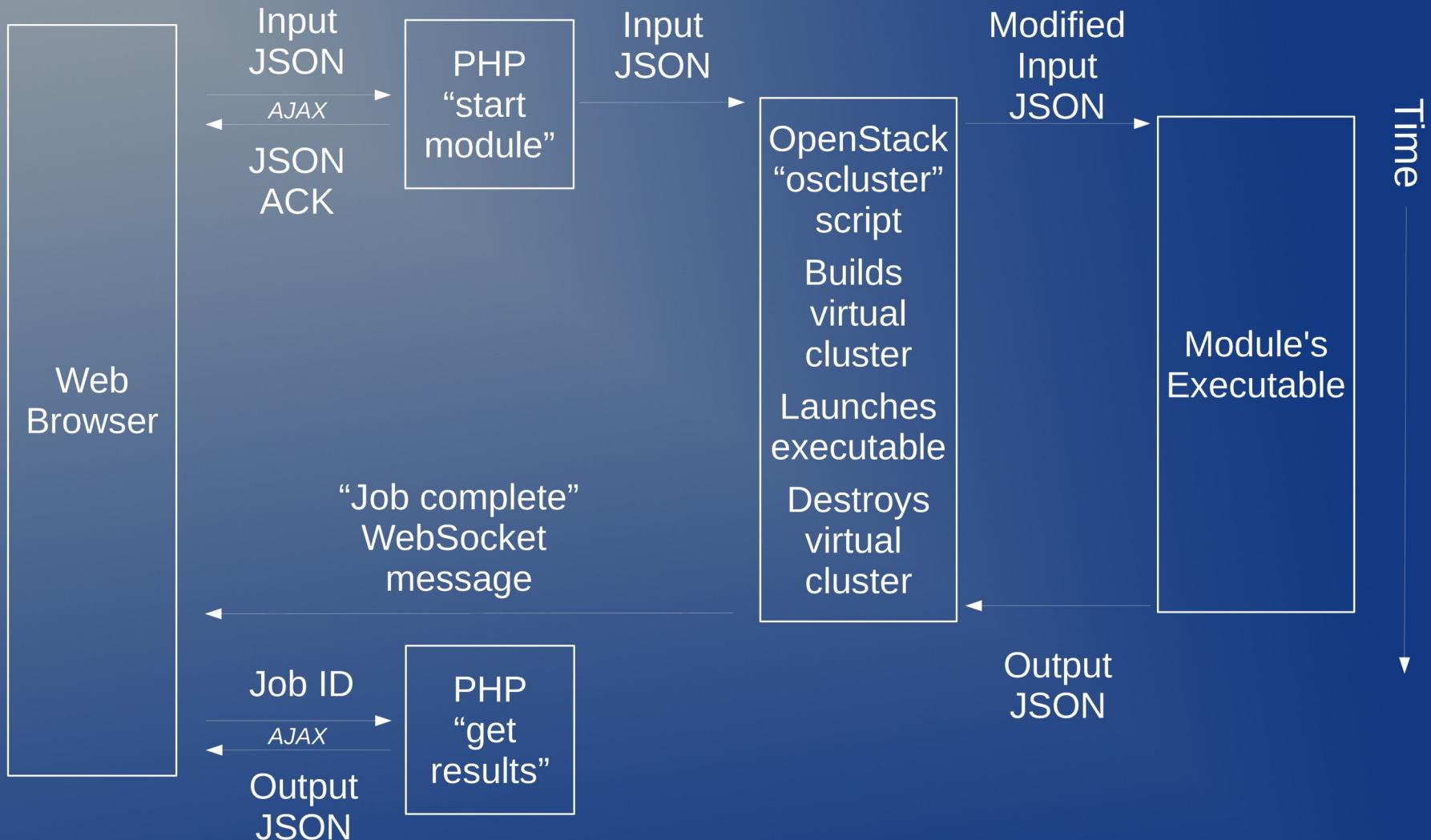
GenApp Integrated with OpenStack Supports Elastic Computing on Jetstream

Public IP
HTTPS
key only SSH

GenApp
web server
VM

Private network
NFS
SSH

virtual cluster node

virtual cluster node

virtual cluster node

virtual cluster node

Storage volume

Emre Brookes and Alexey Savelyev. (2017). PEARC17, ACM.

GenApp Integrated with OpenStack Supports Elastic Computing on Jetstream

Web Browser

Input JSON
AJAX
JSON ACK

PHP "start module"

Input JSON

OpenStack "oscluster" script

Builds virtual cluster

Launches executable

Destroys virtual cluster

Modified Input JSON

Module's Executable

Time

"Job complete" WebSocket message

Output JSON

Job ID
AJAX
Output JSON

PHP "get results"

Emre Brookes and Alexey Savelyev. (2017). PEARC17, ACM.

Outline

- GenApp Background

- **Current**

- Future

Capabilities

- Generation of user interfaces wrapping modules

- Multiple target languages

- Multiple execution models

    - local to remote HPC and cloud

        - Airavata integration

        - OpenStack integration (cloud)

- Messaging

- Users, user management and statistics

- Job management, reattach

- "Cloud" file system

- Caching

- Interactivity. (e.g. repeaters, calculated fields)

- Extensible

    - features added on an as-needed basis

GenApp Apps

In production:

SASSIE-web (J.E. Curtis)
SCT (S. Perkins)

https://sasssie-web.chem.utk.edu/sassie2

300+ users, 9500+ jobs in 2016, 40+ papers

Denfert (J. Perez)
Will it Fit (L. Arleth)
US-SOMO (E. Brookes)
Vortex Shedding (A. Perlstein)
NAMDrunner (A. Savelyev)
QuaFit (F. Spinozzi)

http://genapp.rocks/denfert
http://genapp.rocks/willitfit
http://genapp.rocks/somo
http://genapp.rocks/vortexshedding
http://genapp.rocks/namdrunner
http://genapp.rocks/quafit

In Development:
BioMolAnalysis Suite (A. Savelyev)

Further out:
Bunch (Trewhella)
GenFit (F. Spinozzi)
Memprot (J. Perez)

GenApp Apps : Notes

- SASSIE-web, Joseph E Curtis

  - First generated application, in production

  - Supporting SASSIE-web was the core driver of initial capabilities

    - File systems, job management, …

GenApp Apps : Notes

- Denfert, J. Pérez
  - Created a few hours after receiving the module's executable
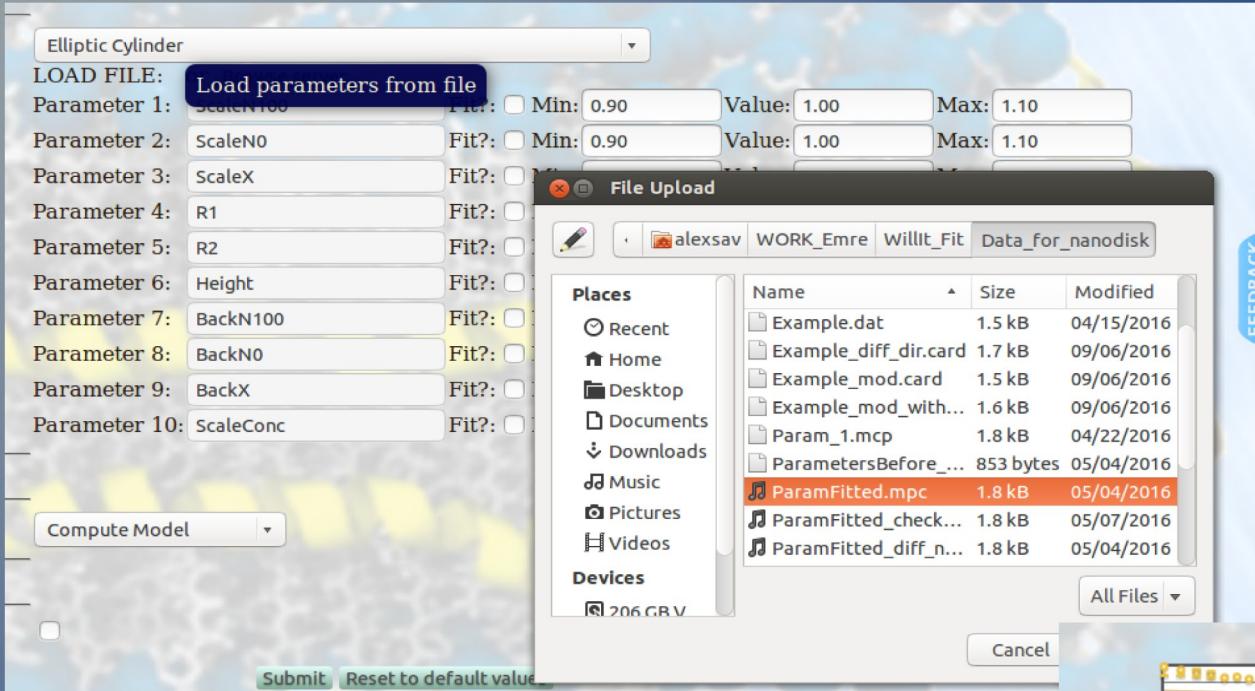
GenApp Apps : Notes

- Vortexshedding, (A. Perlstein)

    - NSF XSEDE ECSS

    - Runs on NSF XSEDE HPC Resource

    - Has driven improvements

        - Detailed job reporting (as req. by XSEDE)

        - Captcha, email validation, caching, video output, ...

    - ECSS lead surprised how quickly the science gateway was up

    - Lead PI expects eventual 10k+ users

GenApp Apps : Notes

- Willitfit (L. Arleth)
    - Created by A. Savelyev
    - Has driven improvements
        - Load parameter file populates fields

WILLITFIT: A Framework for Fitting of Constrained Models to SAS Data

- **Advanced Input:** `lrfile.input`

  - dynamic population of numerous fields while reading from parameter file(s)

- **Advanced Output:** `2Dplot.output`

  - jQuery FLOT package
  - Interactivity: zooming, changing X-, Y-scales; saving to file
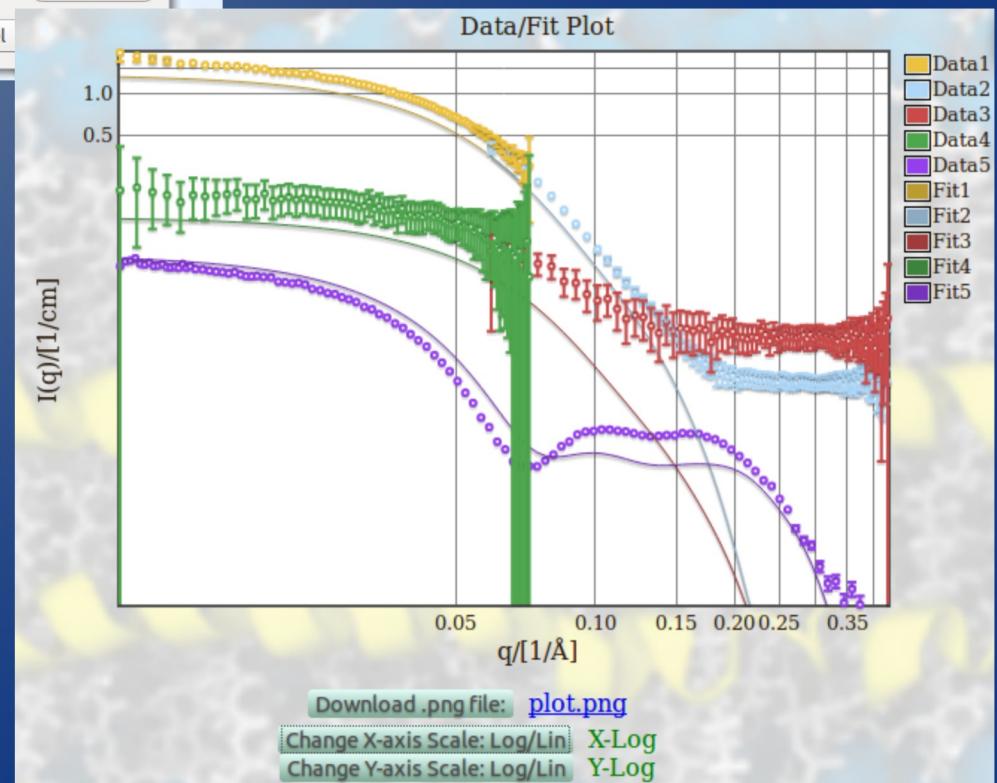
Willt Fit - Beta:

- Underlying code: C++
- Wrapped in PYTHON (~2000 lines)

Pedersen et al. JAC. 46(6), 2013, 1894–1898

Brookes, E. H. CCP-SAS @ Cardiff – 19 June 2017

QUAFIT: A Novel Method for the Quaternary Structure Determination from SAS Data

Input menu for 'QuaFit' module

Output

QuaFit - Beta:

- Underlying code: Fortran
- Wrapped in PERL (~2000 lines)
- Advanced features:
  - nested repeaters (complex layout)
  - interactive output plot [jQuery FLOT]
  - 3D atomic structure [jsMol]

GenApp Apps : Notes

- NAMDrunner (A. Savelyev)

  - Created by A. Savelyev

  - Allows running of NAMD via a GenApp generated gateway

  - Currently running on NSF/XSEDE/Jetstream

  - Has driven improvements

    - OpenStack integration for dynamic creation of virtual clusters

    - Identity management using OAuth2 to allow users to login in with various credentials (XSEDE, Globus, Google)

GenApp Apps : Notes

- New application requirements drive new capabilities
    - Every application can benefit

Resources

- Primary host at University of Tennessee Knoxville

    - "Entropy" server 128 core, 256 GB ram, 8 Tesla K20m GPUs, Rocks OS

    - Running HTML5/PHP

- Indiana University Quarry nodes (moving to Jetstream)

    - Trac wiki with integrated subversion repository

- XSEDE Jetstream

    - XSEDE ECSS vortexshedding gateway

    - SASSIE-web instance

    - NAMDrunner gateway

- AWS

- XSEDE TG-MCB140255 *Computational support for small angle scattering for advanced analyses of structural data in chemical biology and soft condensed matter*

- ORNL Titan, UK SCARF (pending)

Resources – Personnel

- Alexey Savelyev, Post Doc.

  - Framework developer

  - Application developer

  - Module wrapper

  - US-SOMO

  - WillItFit

  - Quafit

- Emre Brookes

  - Primary developer

- External Application/Module developers

  - Joseph Curtis, PI and personnel

    - SASSIE

  - David Wright

    - SCT

  - Arne Perlstein, PI's assigned personnel (currently Josef Sabuda)

    - Vortexshedding

Outline

- GenApp Background

- Current

- **Future**

Future

- Ongoing
  - More application & module wrapping
    - Training others to wrap
      (and making it as easy as possible)
    - We could wrap a lot given sufficient resources
  - Easy install of applications for setup of web-servers and standalone
- Future
  - Instance generator / VM's and/or simple JSON
  - Module portal
  - Target for API access
  - Containerization
  - Apache membership
  - Automatic regression testing

Pubs

- Brookes, E H, 2014, *An Open Extensible Multi-Target Application Generation Tool for Simple Rapid Deployment of Multi-Scale Scientific Codes,* ACM, 10.1145/2616498.2616560

- Brookes, E H, Anjum, N, Curtis, J E, Marru, S, Singh, R, Pierce, M, 2014, *GenApp module execution and airavata integration,* IEEE Press, 10.1109/GCE.2014.12

- Brookes, E H, Anjum, N, Curtis, J E, Marru, S, Singh, R, Pierce, M, 2015, *The GenApp framework integrated with Airavata for managed compute resource submissions*, Concurrency and Computation: Practice and Experience, Wiley & Sons, 10.1002/cpe.3519

- Wright DW, Nan R, Hui G, Curtis JE, Brookes EH, Perkins SJ. CCP-SAS - Novel Approaches for the Atomistic Modeling of Small Angle Scattering Data in Biology. Biophysical journal. 2015 January 27; 108(2):191a.

- Brookes EH, Kapoor A, Patra P, Marru S, Singh R, Pierce M. GSoC 2015 student contributions to GenApp and Airavata. Concurrency and Computation: Practice and Experience. 2015 October 31 DOI: 10.1002/cpe.3689

- Emre Brookes and Alexey Savelyev. (2017). GenApp Integrated with OpenStack Supports Elastic Computing on Jetstream. PEARC17, ACM.

- Alexey Savelyev and Emre Brookes (2017) GenApp: Extensible Tool for Rapid Generation of Web and Native GUI Applications, J. Future Generation Computer Systems, (invited, submitted)

- Emre Brookes and Alexey Savelyev. (2017). Architecture and Performance of a GenApp Generated NAMD Science Gateway on Jetstream. SuperComputing, ACM (submitted)

Thanks for listening

*Questions: ask now or
email me at emre@biochem.uthscsa.edu*

Acknowledgments

- Joseph Curtis

- Alexey Savelev

- To E. Brookes

    - NIH NIGMS K25GM090154 / PI

    - NSF CHE-1265817 / PI

- NSF XSEDE TG-MCB140255

- UTHSCSA BCF

- To N. Anjum, A. Kapoor, P. Patra

    - Google Summer of Code

SCHOOL OF MEDICINE
UT HEALTH SCIENCE CENTER®
DEPARTMENT OF BIOCHEMISTRY
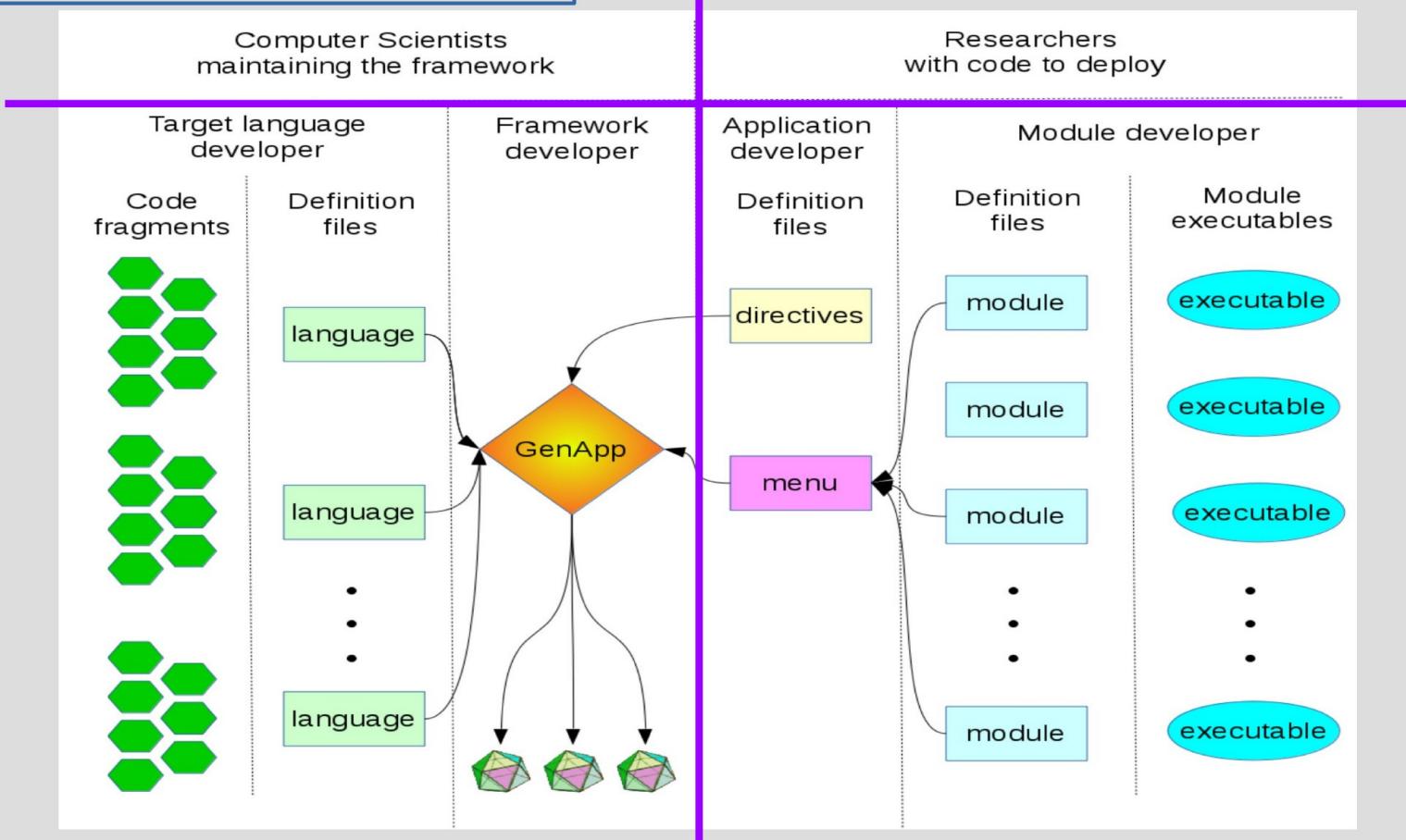
Encountering GenApp: Extending Horizon

Cool stuff:

- Modifying fragments of code
- Adding new /Extending types
- Conditional code generation
- Browser compatibility
- Admin utilities
- ….
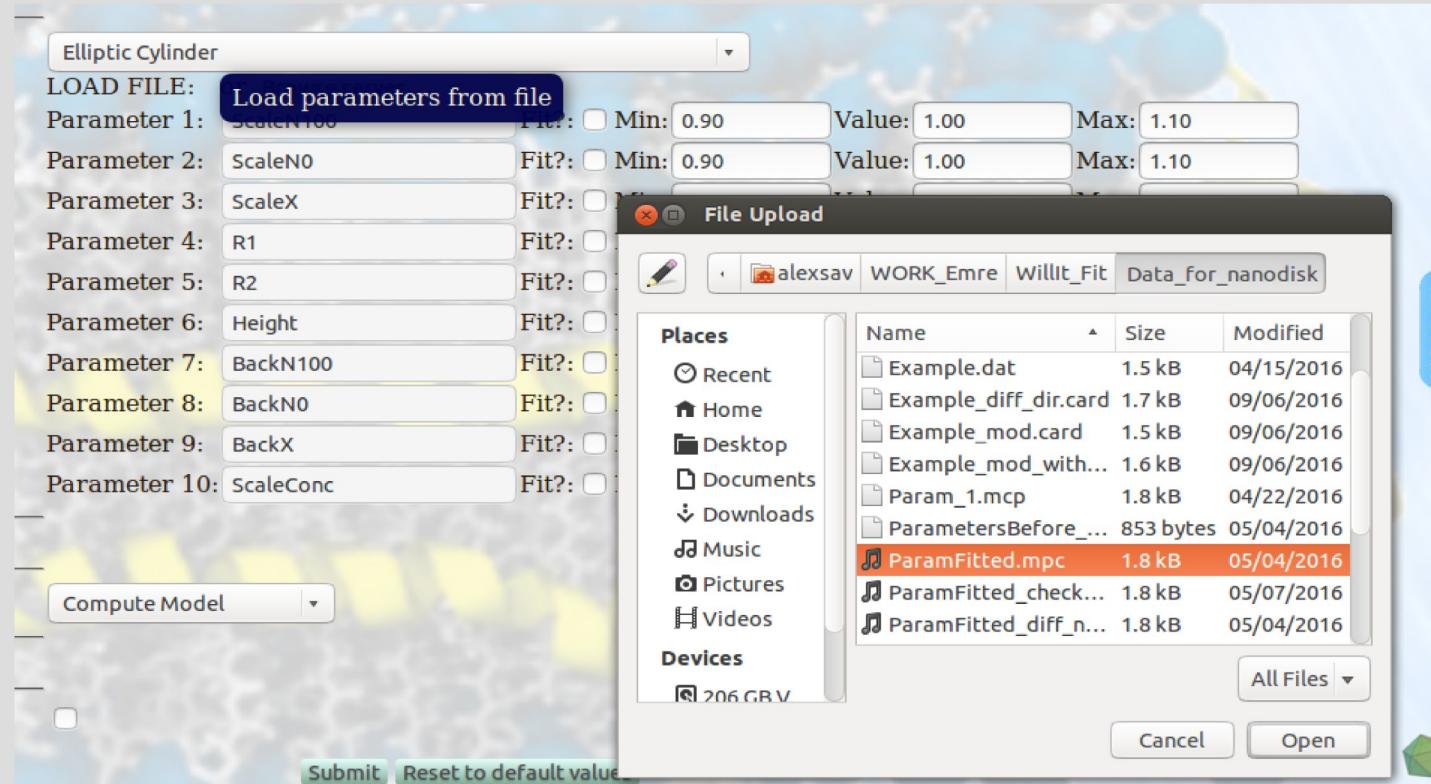
Web Development:

HTML5
CSS
JavaScript (jQuery)
PHP
MongoDB

Computer Scientists maintaining the framework

Researchers with code to deploy

Target language developer	Framework developer	Application developer	Module developer

Code fragments — Definition files — language

GenApp

Definition files — directives — module — executable

Definition files — menu — module — executable

Module executables — executable

language

language

Advanced "GenApping": *WillItFit (1)*
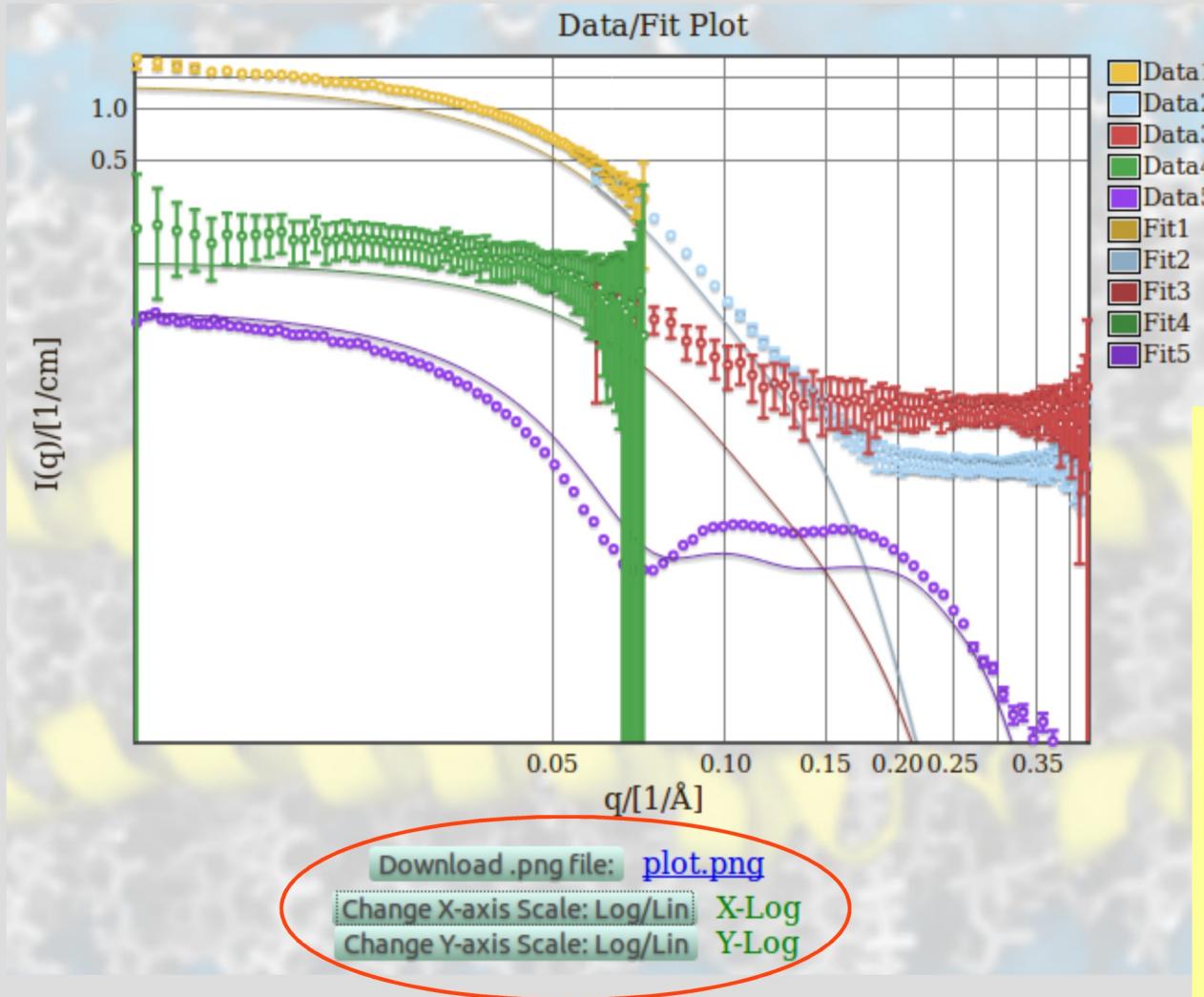
- Uploading parameter file, `lrfile.input` (numerical values, whitespace_separated etc.)


```
{
  "role"    :    "input",
  "id"      :    "param_choice_file_anis_core_shell_micelles",
  "label" :    "LOAD FILE: ",
  "type"   :    "lrfile",
  "setinputfromfile"    : "whitespaceseparated_reverselogic",
  "setinputfromfileids": "..list_of_ids..",
  "repeat":    "model_list_box:anis_core_shell_micelles",
  "help"   :    "Load parameters from file"
}
```

Advanced "GenApping": *WillItFit (2)*

- Advanced `Plot2D.output` options (save to file, change X-, Y-axis scales independently)

Data/Fit Plot

jQuery FLOT package
http://www.flotchart.org

```
{
    "role"   : "output",
    "id"     : "fit_plot",
    "label"  : "Fit Plot:",
    "type"   : "plot2d",
    "height" : "400px",
    "width"  : "700px",
    "pan"    : "false",
    "zoom"   : "false",
    "backgroundcolor"  : "white",
    "selzoom"          : "true",
    "changescalex"     : "true",
    "changescaley"     : "true",
    "savetofile"       : "true",
    "rotatedylabel"    : "true",
    "hover"            : "true"
}
```

Advanced "GenApping": MULCh *(2)*
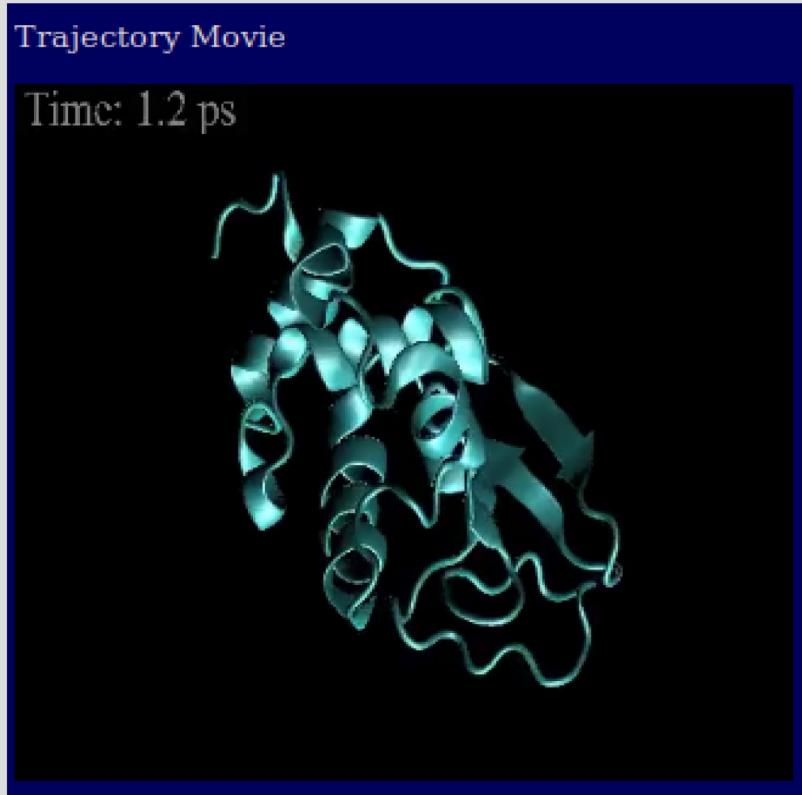
- Calculated fields (inter-dependent numerical fields)


```
{
    "role"     : "input",
    "id"       : "delta_rho_1",
    "label"    : "Delta rho",
    "type"     : "text",
    "repeat"   : "contrast_points",
    "required" : "true",
    "calc"     : "d_rho1*fdo2 + fdo2_1",
    "pattern"  : "^(-)?((([1-9][0-9]*)|(0))?([.][0-9]+)?([eE][-+]?[0-9]+)?$"
}
```
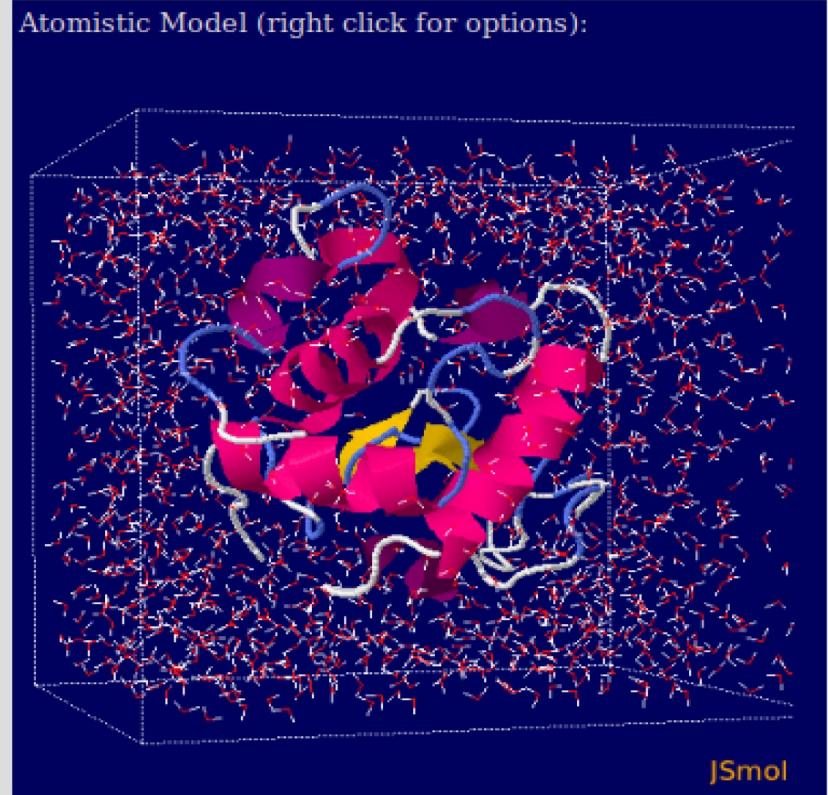
Advanced "GenApping": ParamMD *(1,2)*

- Trajectory Movie generation [.mp4, .webm]
`video.output`

Trajectory Movie
Time: 1.2 ps

```
{
   "role"   : "output",
   "id"     : "vid",
   "type"   : "video",
   "label"  : "Trajectory Movie",
   "width"  : 400,
   "height" : 450
}
```

- Structure visualization, JSmol
`atomicstructure.output`

Atomistic Model (right click for options):

JSmol

```
{
   "role"     : "output",
   "id"       : "outputpdb_view",
   "label"    : "Atomistic Model",
   "type"     : "atomicstructure",
   "jsmoladd" : "hide HOH;spin on",
   "width"    : 450,
   "height"   : 450
}
```

Advanced "GenApping": Admin Utilities

- Job History, jobs information within specified time frame (PHP, MongoDB)

NAMDrun - Beta 0.1

Logoff alexey
Help on

Job monitor   Integrity check   Users   User management   Job history

Start Date 2016-10-10

End Date 2016-11-03

Submit   Reset to default values

name	email	duration (h)	running	finished	cancelled
Totals		102.411	1	221	3
alexey	alexsav.science@gmail.com	1.547	1	99	1
amirayuyue	amirayuyue@gmail.com	0	0	0	0
cpayne	christy.payne@uky.edu	0	0	0	0
danielma	danmart_us@yahoo.com	0	0	0	0
emre	emre@biochem.uthscsa.edu	0.292	0	49	0
graceb	gracebrannigan@gmail.com	0	0	0	0
gumbart	gumbart@physics.gatech.edu	0	0	0	0
hwang	hhwang8@gatech.edu	56.094	0	65	2
jvermaas	joshua.vermaas@nrel.gov	44.476	0	8	0
mocohen	mocohen@uchicago.edu	0	0	0	0
pcardena	pacl3@gatech.edu	0	0	0	0
ttjoseph	thomas.joseph@uphs.upenn.edu	0	0	0	0

genapp.rocks

FEEDBACK