TimeW	Title	Presenter
Day 1	Programming & The C Language	
Day 2	Parallel Programming: MPI and OpenMP	
Day 3	Abstraction, Struct & C++	
Day 4	UI & Qt	Frank
9:00-9:20	Setting Up For Today	
9:30-10:30	SimCenter	Frank
10:30-11:30	Exercise uqFEM	Frank
11:30-12:30	DesignSafe Website - Interactive	You
12:30-1:15	LUNCH	
1:15-3:00	Agave Interactive	
3:00-3:30	Wrap Up - Discussion	Frank


SimCenter
enter
NSF
NHERI
Berkeley
UNIVERSITY OF CALIFORNIA

We Are a Virtual EF

We are producing **software applications** and **educational activities** to advance research in NHE.

Leadership Group

Ahsan Kareem
Notre Dame

Laura Lowes
Washington

Greg Deierlein
Stanford

Sanjay Govindjee
UC Berkeley

Camille Crittenden
UC Berkeley

Matt Schoettler
UC Berkeley

Frank McKenna
UC Berkeley

Postdoctoral Team

Nikhil Padhye
UC Berkeley

Chaofeng "Charles" Wang
UC Berkeley

Wael Elhaddad
UC Berkeley

Adam Zsarnoczay
Stanford

Michael Gardner
UC Berkeley

Peter Sempolinski
Notre Dame

Faculty Involved

- UW Pedro Arduino, **P. Mackenzie-Helnwein**, Michael Motley
- UCB Jonathan Bray, Filip Filippou, Paul Waddell
- UCSD Joel Conte
- USC Ewa Deelman, Patrick Lynett
- GSU Ann-Margaret Esnard
- JHU Judy Mitrani-Reiser
- ND Tracy Kijewski-Correa, Alex Taflanidis
- Stanford Kincho Law, Eduardo Miranda, Jack Baker
- U Memphis Ricardo Taborda
- UCLA Ertugrul Taciroglu
- CSU L. Beach Vesna Terzic
- GT Iris Tien
- Columbia George Deodatis

Software Experts Say We Need to Solve Grand Challenges related to Natural Hazards

1) Applications that generate UQ in Response Quantities
2) Applications to perform Performance Based Engineering
3) Applications for Community Resiliency
4) Educational Applications

Software Products We Are Developing

- **W**e are building a number of research applications:
  - **uqFEM:** : To enhance FEM applications with UQ & Optimization
  - **EE-UQ**: To provide response of buildings to earthquake events
  - **CWE-UQ**: To provide response of buildings to wind events
  - **PBE**: EE-UQ + CWE-UQ plus Downtime and Loss estimation
  - **RDT**: *To estimate* Regional Resiliency given Multiple Hazards
  - **OTHER**

So How Are We Doing It?

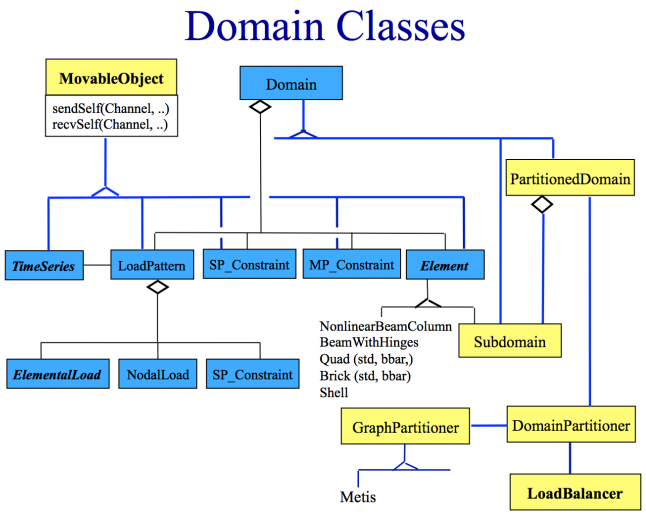
SimCenter is developing an **application Framework** that will Enable creation of Scientific Workflow Applications for researchers working in field of NHE

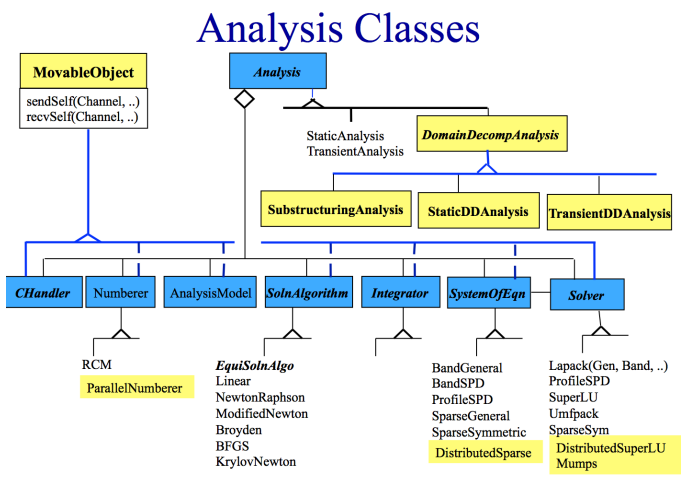
We are Developing **Interfaces, Code to meet the Interfaces** & **Applications.** The applications are being designed to be **flexible** and **extensible.**

Definition

Application Framework: An application framework is a collection of software for building applications in a specific domain. **The framework defines the interfaces between the components** of the software, **provides code that implements some components, and finally provides example applications** that can be developed using the provided software.

OpenSees is an example Framework.





Released originally with 3 elements and about 4 simple materials. Now over hundreds of both

Definition

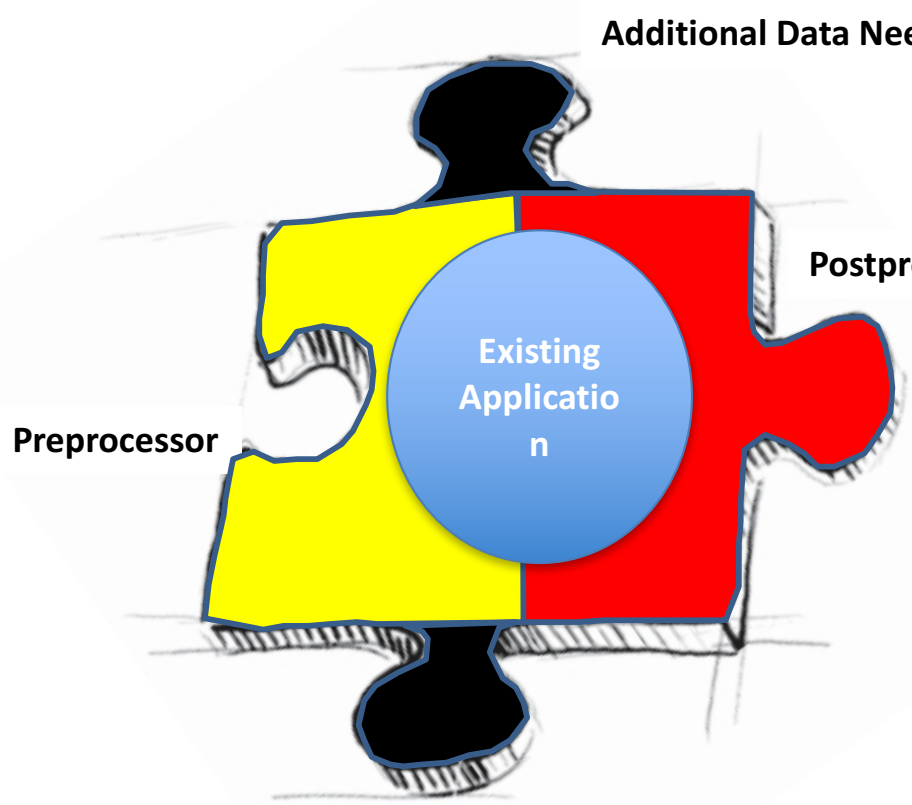
Scientific Workflow Application: A scientific workflow is the **automation** of a process in which information is passed from **one application to the next**.

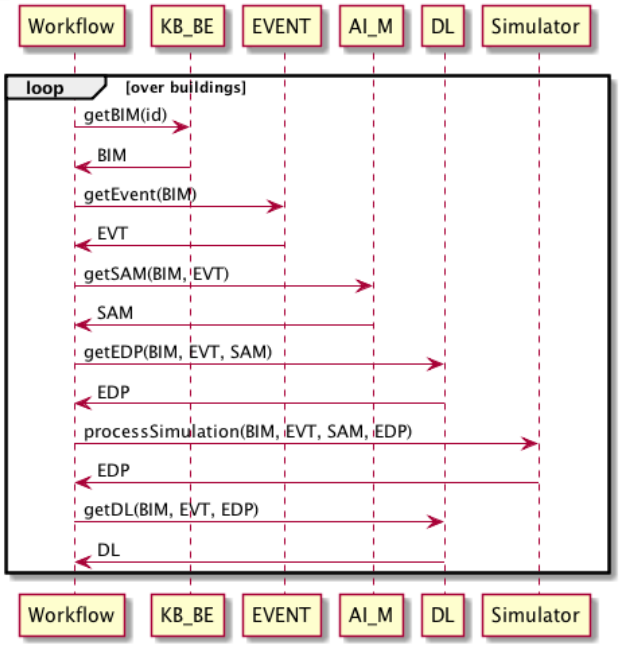
SimCenter classes
are
Applications

Existing Applications, which of course do not work together

SimCenter defining interfaces they must meet!

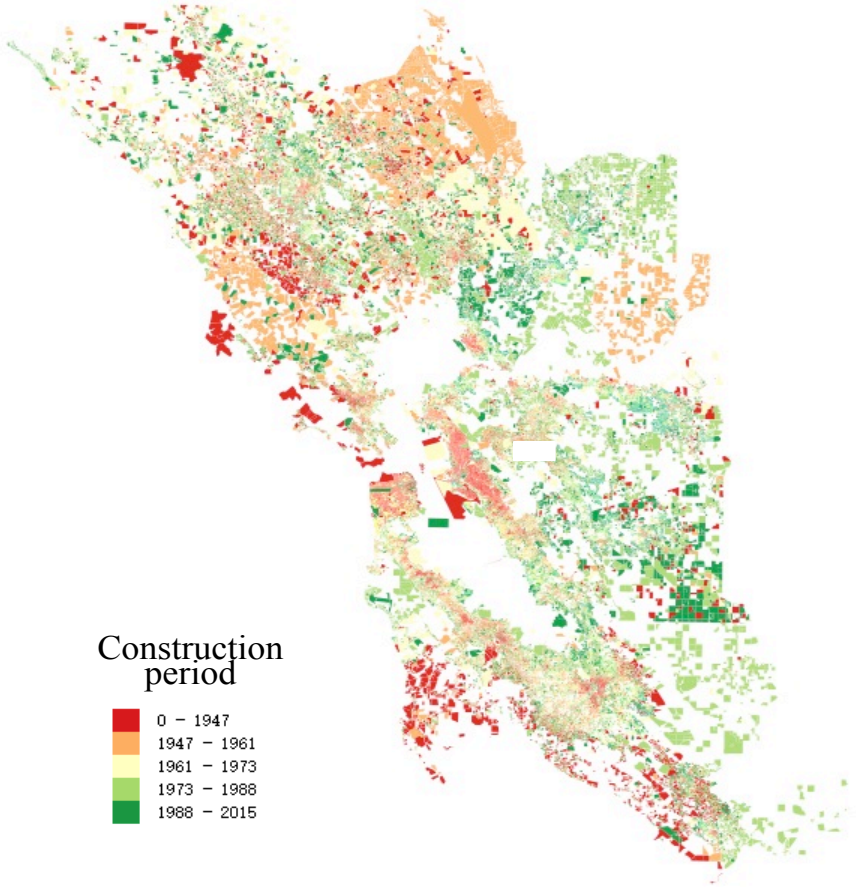
And Writing Code to incorporate Existing Applications into Workflow

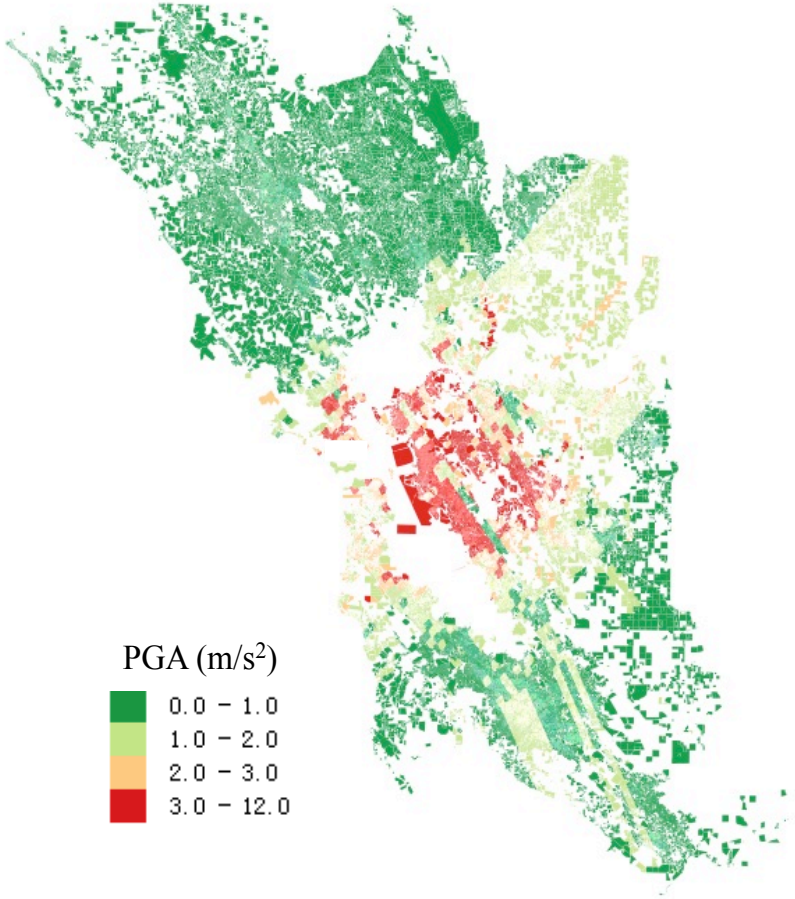




One SimCenter Application:
Workflow for Regional Earthquake Response

Input Data



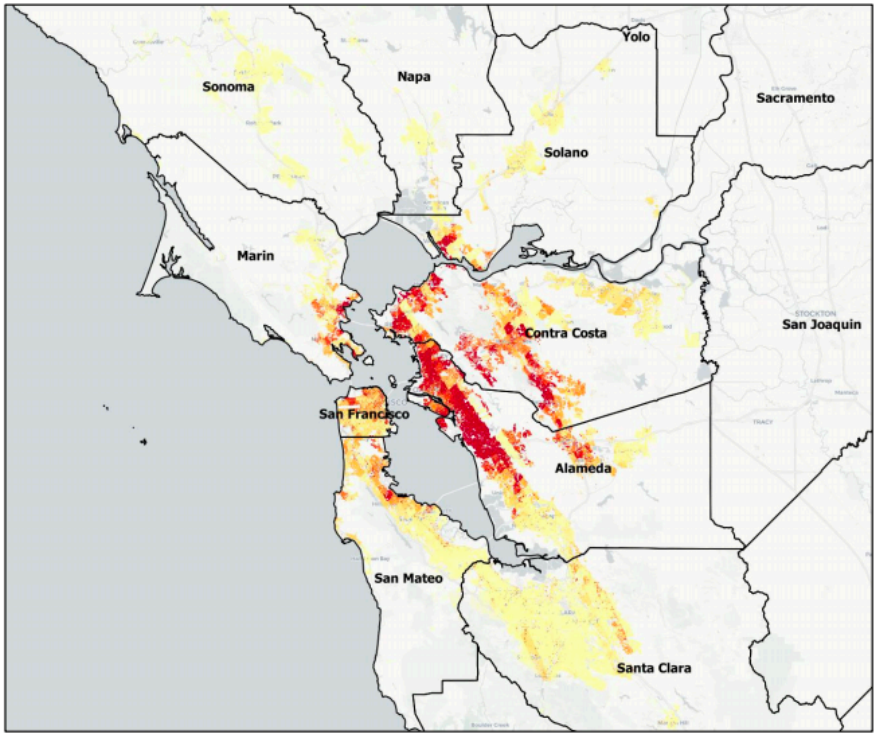


UrbanSim

Data science, simulation and visualization to learn from the past, inform the present, and shape the future of communities everywhere.

SW4

Output



SimCenter Workflow

Input (JSON file)

```
workflow >more workflow1.json
{
   "Name": "Workflow 1",
   "Author": "fmk",
   "WorkflowType": "Regional Simulation",
   "buildingFile":"buildings.json",
   "Applications": {
      "Buildings": {
         "BuildingApplication": "UrbanSimDatabase",
         "ApplicationData": {
            "Min": "1",
            "Max":"1800000",
            "parcelsFile":"../NHERI/parcels.csv",
            "buildingsFile":"../NHERI/buildings.csv"
         }
      },
      "Events": [
         {
            "EventClassification": "Earthquake",
            "EventApplication": "LLNL-SW4",
            "ApplicationData": {
               "pathSW4results": "../NHERI/Hayward7.0/",
               "filenameHFmeta":"../createEVENT/HFmeta"
            }
         }
      ],
      "Modeling": {
            "ModelingApplication": "MDOF-LU",
            "ApplicationData": {
               "hazusData":"../createSAM/data/HazusData.txt"
            }
```

"Applications" is a list of applications

Each Application is a JSON object with specific data

App specific Data

To Change the Workflow:

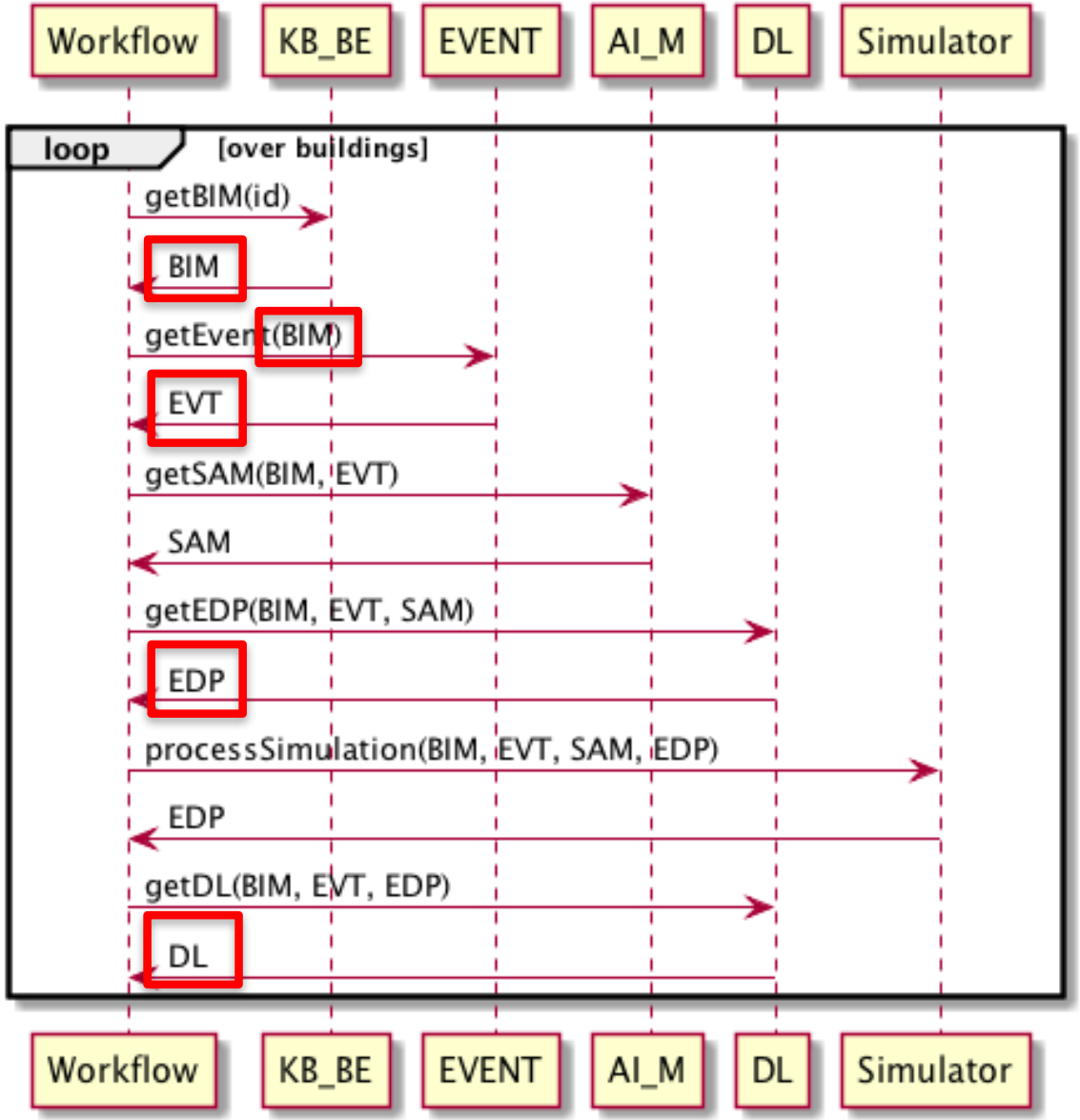
```
"Events": [
   {
      "EventClassification": "Earthquake",
      "EventApplication": "LLNL-SW4",
      "ApplicationData": {
         "pathSW4results": "/Users/fmckenna/NHERI/Hayward7.0/",
         "filenameHFmeta":"/Users/fmckenna/NHERI/Workflow1.1/createEVENT/HFmeta"
      }
   }
],
"Modeling": {
```

```
},
"Events": [
   {
      "EventClassification": "Earthquake",
      "EventApplication": "SHA-GM",
      "ApplicationData": {
         "scenarioConfig": "./HayWired7.25.json"
      }
   }
],
"Modeling": {
```

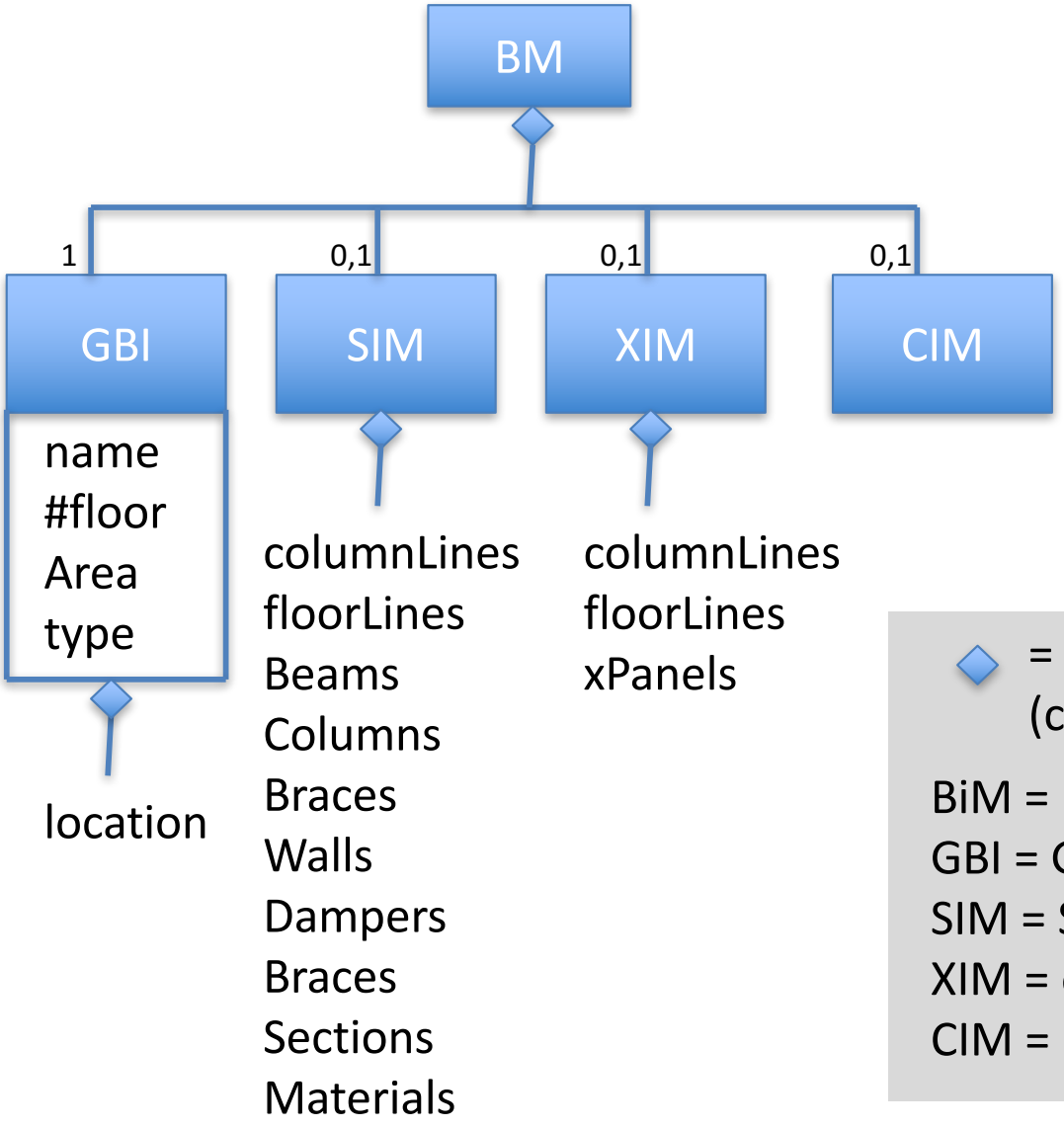
How To Introduce new Applications - config file the applications loads

```
"EventApplications": {
  "API": {
    "Inputs": [
    {
        "id": "filenameBIM",
        "type": "string",
        "description": "name of bim file",
        "default": "BIM.json"
    }
    ],
    "Outputs": [
    {
        "id": "filenameEVENT",
        "type": "string",
        "description": "name of file containing the event data",
        "default": "EVENT.json"
    }
    ]
  },
  "Applications": [
    {
      "Name": "LLNL-SW4",
      "ExecutablePath": "../createEVENT/LLNL_SW4",
      "ApplicationSpecificInputs": [
              { "id":"pathSW4results",
                "type":"string",
                "description":"path to directory containig output point files"
              },
              { "id":"filenameHFmeta",
                "type":"string",
                "description":"path to file containing loacation information on each ouput point file"
              }
      ]
    },
    {
      "Name": "SHA-GM",
      "ExecutablePath": "../createEVENT/SHA-GM.py",
      "ApplicationSpecificInputs":[{
          "id": "scenarioConfig",
          "type": "path",
          "description":"Configuration file of the earthquake scenario including rupture, site grid, GMPE, IM and recor
"
        }
      ]
    }
  ]
},
```

Sequence Diagram for Workflow no UQ



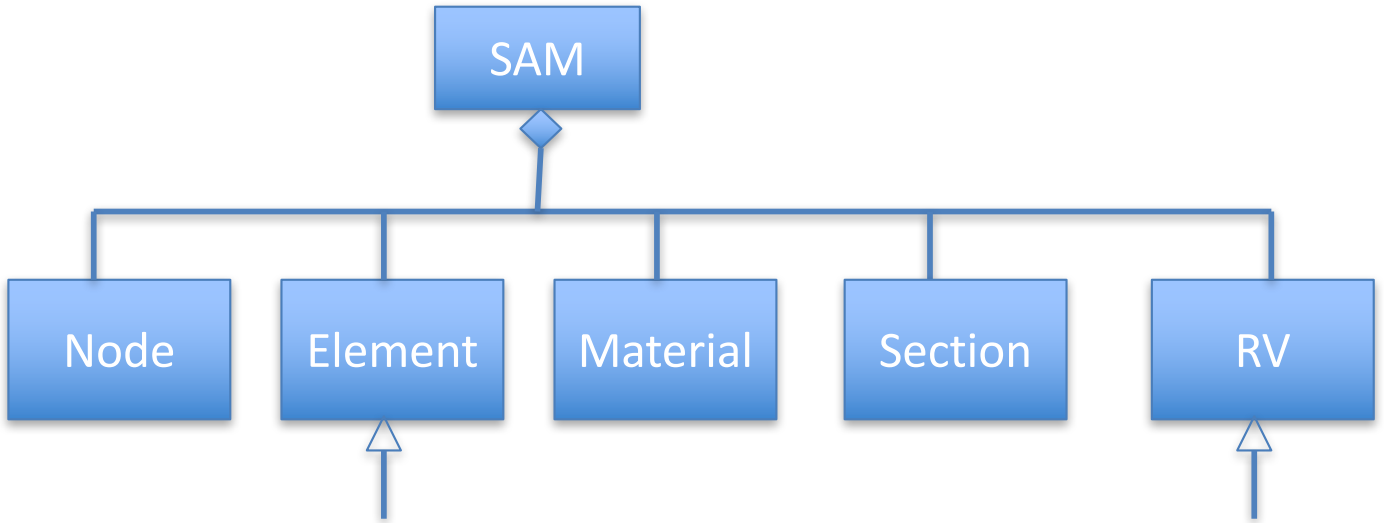
Building Information Model (BIM)



= composition (composed of)

BiM = Building Info Model
GBI = General Building Info
SIM = Structural Info Model
XIM = eXternal Info Model
CIM = Contents Info Model

SAM –Structural Analysis Model

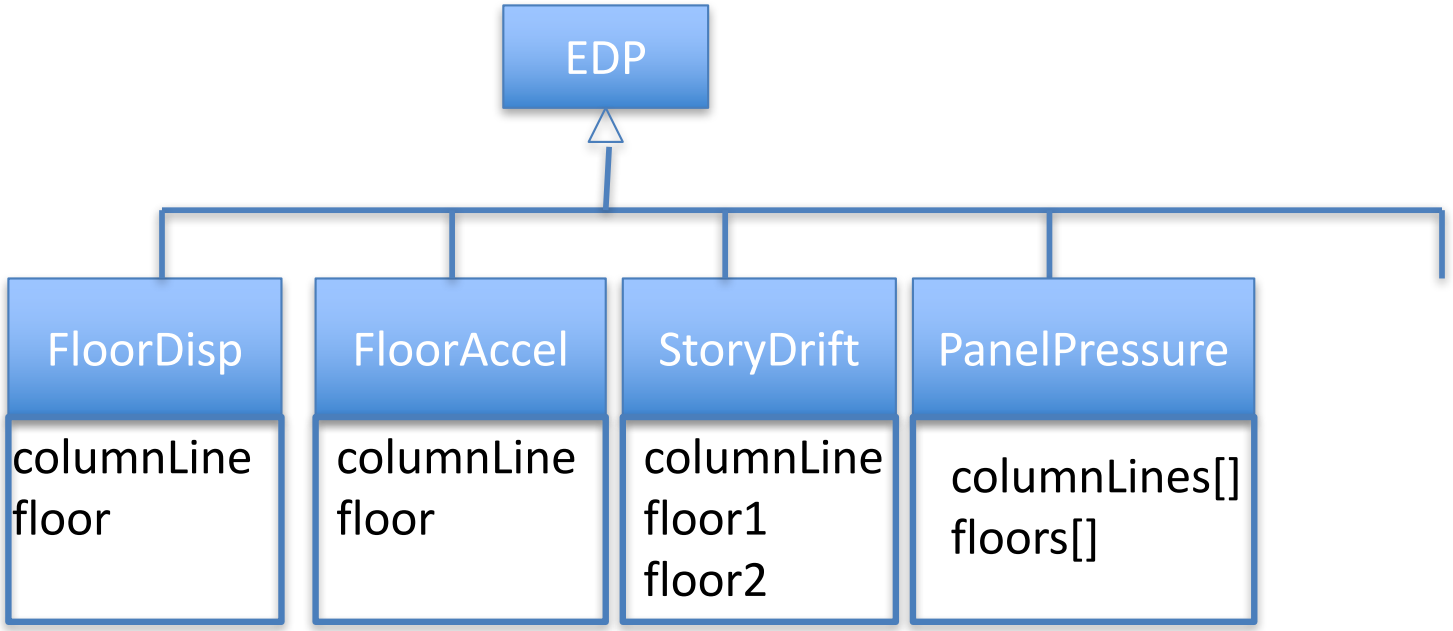


Shell
Quad
ForceBeamColumn
ConcentratedHingeBeamColumn
…

Normal
Lognormal
Weibull
…

◆ = composition
△ = inheritance

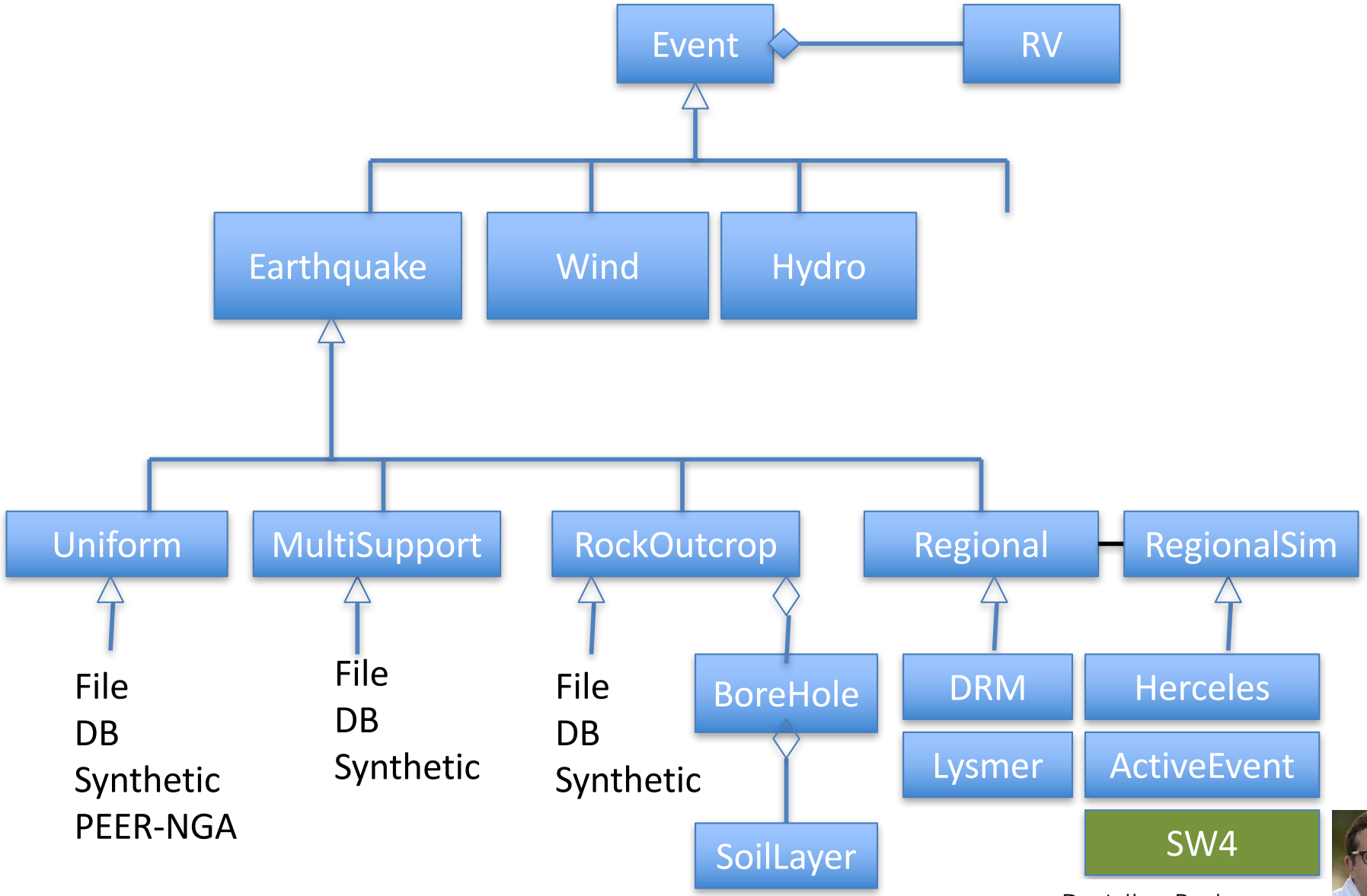
EDP – Engineering Demand Parameters



△ = Inheritance
(is a)

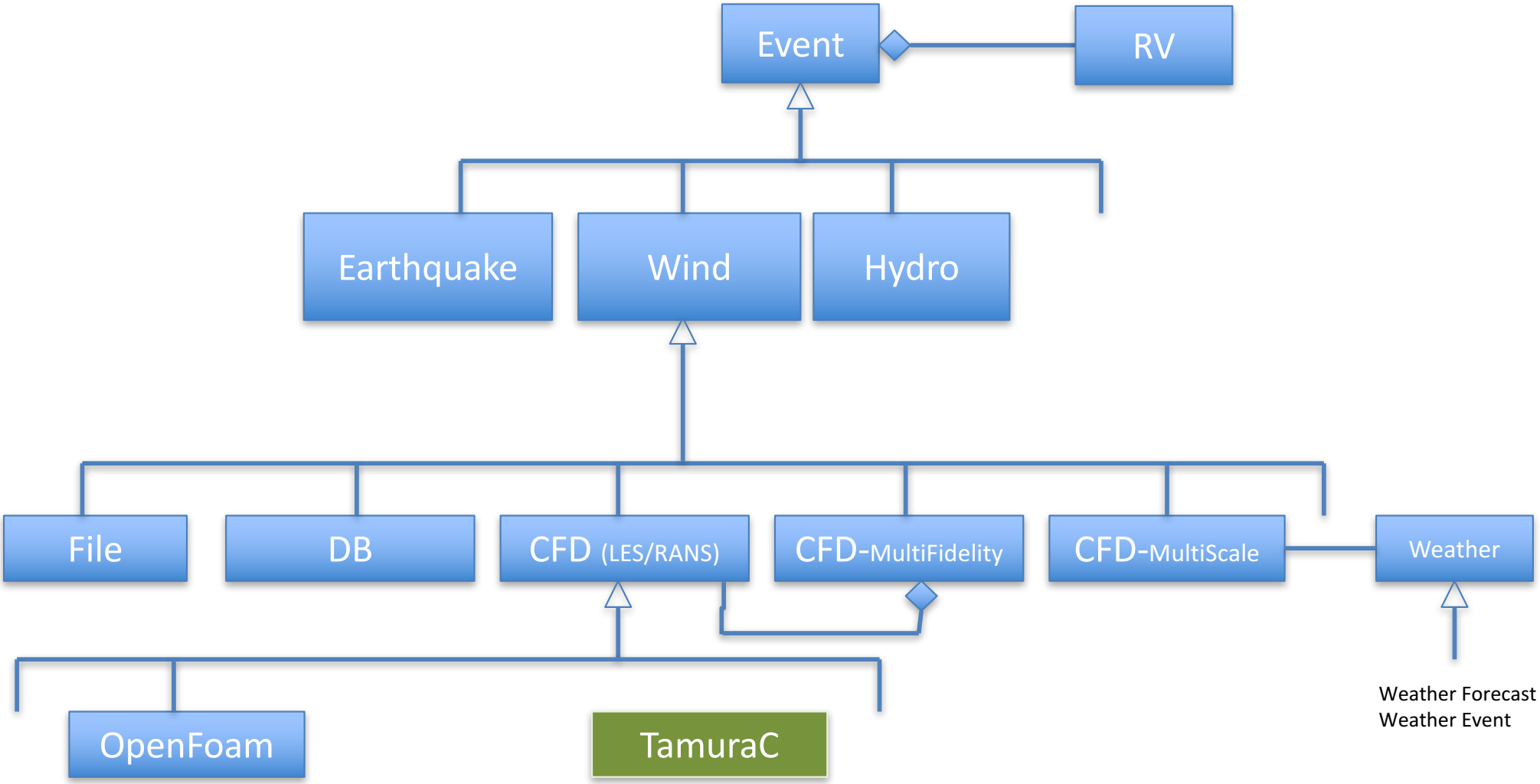
EDP = Engineering Demand Parameter

Event (boundary condition & forces for model)

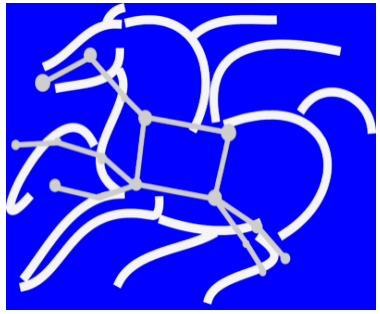


Dr. Arthur Rodgers,
Lawrence Livermore National Lab

Event (boundary condition & forces for model)



Application(s) to Run



```
------------------------------------------------------------------------------
Type           Succeeded Failed  Incomplete  Total     Retries   Total+Retries
Tasks          9297314   0       0           9297314   5417      9302731
Jobs           128539    0       0           128539    1494      130033
Sub-Workflows  38        0       0           38        0         38
------------------------------------------------------------------------------

Workflow wall time                                   : 1 day, 2 hrs
Cumulative job wall time                             : 47 days, 2 hrs
Cumulative job badput wall time                      : 38 secs
```

Research Tools Release Schedule

uqFEM

- V1.0 (June 2018)
- V2.0 (2019)
- V3.0 (2020)
- V4.0 (2021)

CWE-UQ

- V1.0 (June 2018) Bluff Body
- V2.0 (Sept 2018) Building
- V3.0 (2019) UQ

AI

EE-UQ

- V1.0 (Aug 2018) Uniform
- V2.0 (Sept 2018) Rock Outcrop
- V3.0 (2019) Soil Box

AI

PBE

- V1.0 (Sept 2018) Earthquake
- V2.0 (2019) Wind
- V3.0 (2020) Water

AI

Workflow Application Release Schedule

RDT

- V1.0 (2019) Earthquake
- V2.0 (2020) Wind
- V3.0 (2021) Water

Regional Earthquake Response

- V1.0 (June 2018)
- V2.0 (2019)
- V3.0 (2020)

Water

- V1.0 (Sept 2018)
- V2.0 (2020)
- V3.0 (2021)

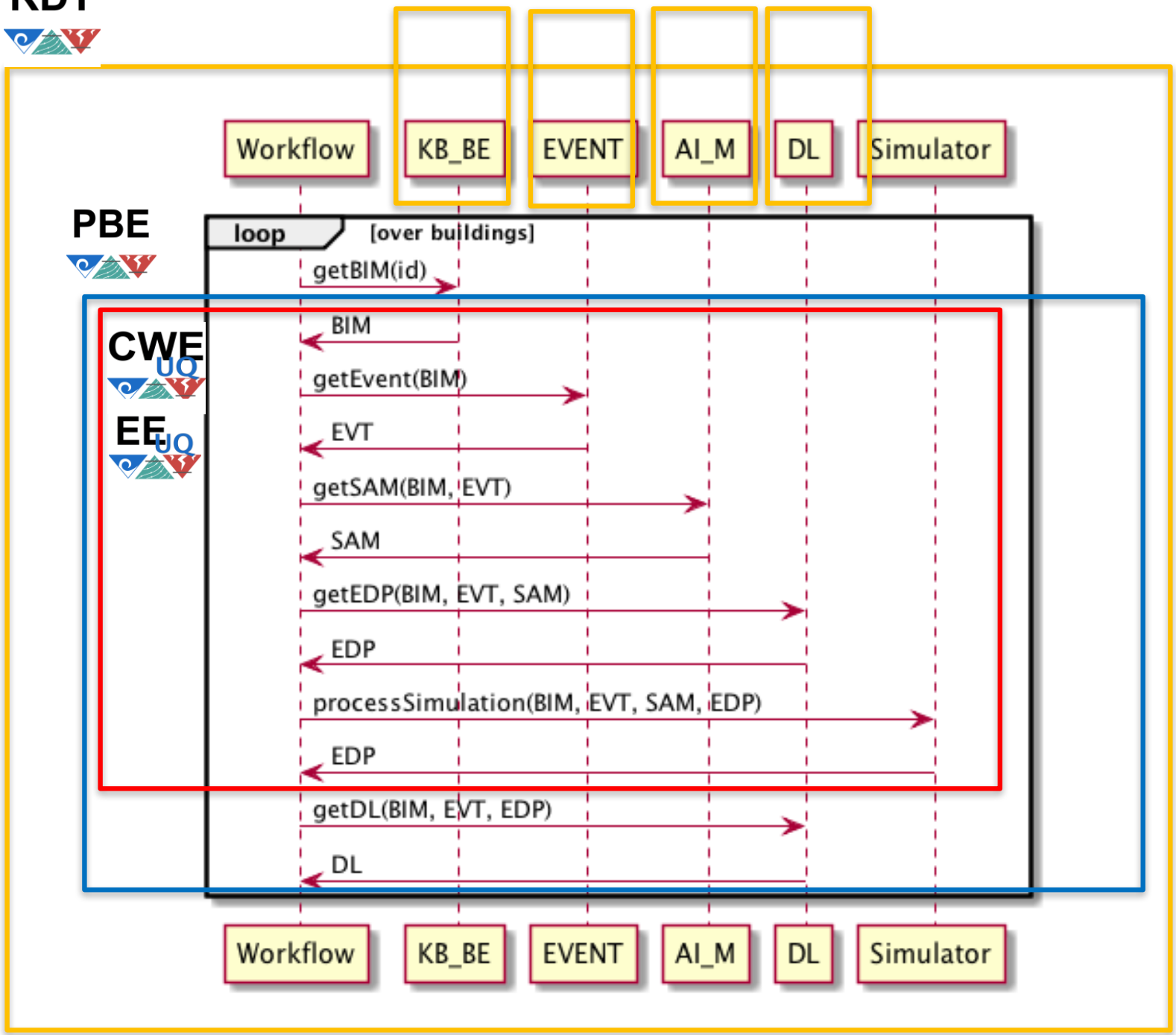
Hurricane

- V1.0 (2020)
- V2.0 (2021)

MultiHazard

- V1.0 (2021)

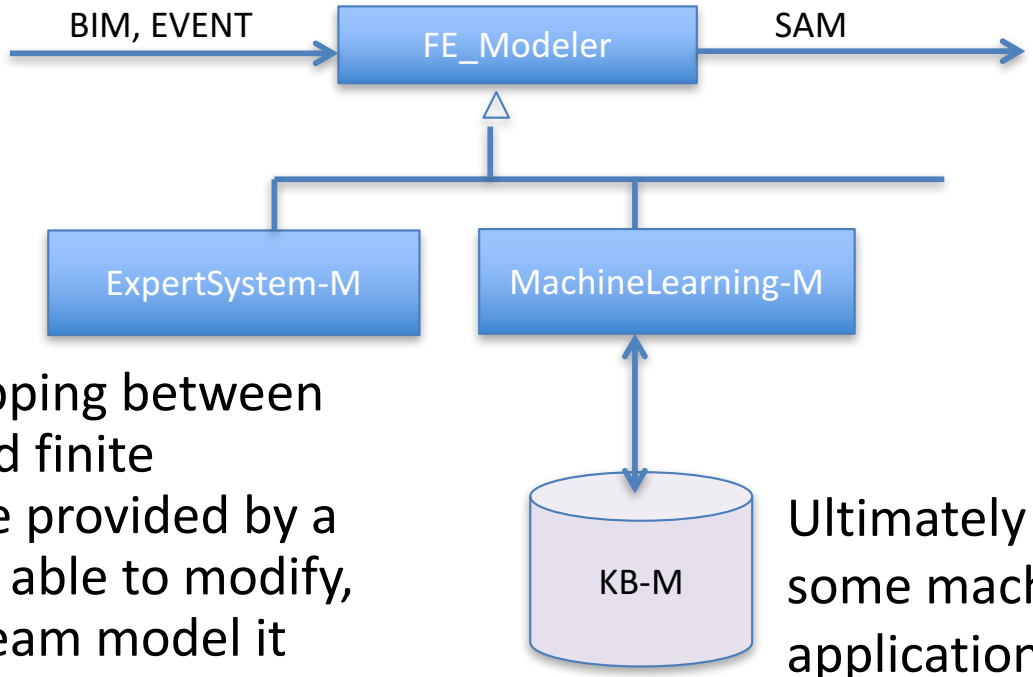
each application developed is also an independent application!



AI use in SimCenter Products

1. Given BIM provide SAM
2. Provide BIM Information given a Region

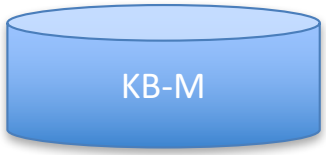
AI produces the SAM (structural analysis) models.



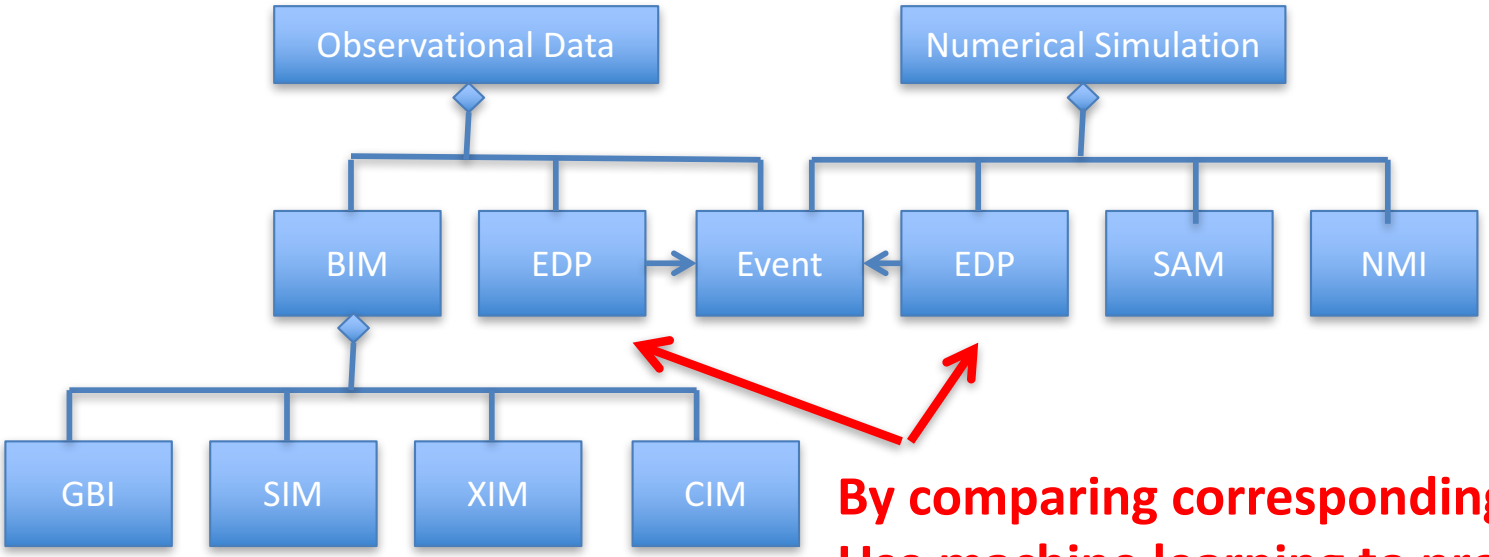
Initially the mapping between components and finite elements will be provided by a file, user will be able to modify, e.g. **if** a steel beam model it with N elements of type X **else if** concrete beam model with M elements of type Y

Ultimately will also provide some machine learning application that will use a knowledge base, KB-M, for the data.

KB-M



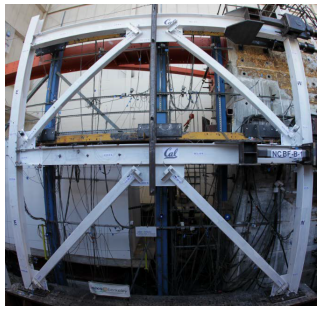
Knowledge base contains information on **observed data** (experimental and actual buildings) and **numerical simulations**.

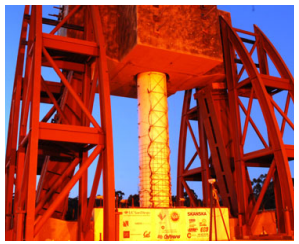


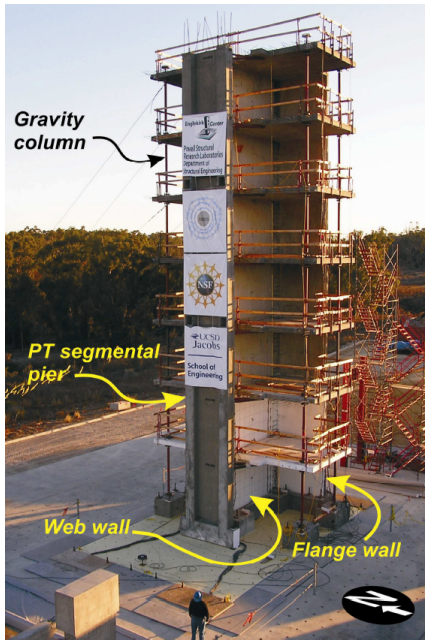
By comparing corresponding responses
Use machine learning to provide SAM (multi model with RV for each model) given new BIM.

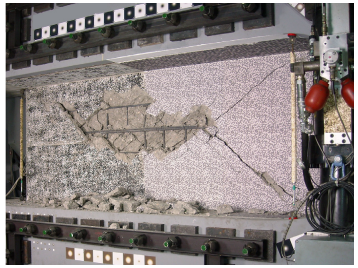
Collecting Experimental Datasets :

Dataset Name	Hazard	ndm	Material	Structural Type	Year	Laboratory	FE Models	# Models	# Random Models
18-Story Steel Hi-Rise at E-Defense	Seismic	3D	Steel	Moment Frame	2013	E-Defense	OpenSees	4	200
2-Story NCBF at UCB	Seismic	2D	Steel	Braced Frame	2012	UC Berkeley	OpenSees	3	0
7-Story RC Building at UCSD	Seismic	3D	RC	Shear Wall	2006	UC San Diego	Various	0	0
Collapse of 4-Story Steel Building at E-Defense	Seismic	3D	Steel	Moment Frame	2007	E-Defense	OpenSees	1	0
Collapse of RC Frame at Tsinghua University	Seismic	2D	RC	Moment Frame	2011	Tsinghua University	OpenSees, MSC Marc	4	200
RC Columns at Tsinghua University	Seismic	2D	RC	Moment Frame Comp.	2011	Tsinghua University	OpenSees, MSC Marc	6	200
RC Joint at Tsinghua University	Seismic	2D	RC	Moment Frame Comp.	2011	Tsinghua University	OpenSees, MSC Marc	4	0
Large-Scale Bridge Column at UCSD	Seismic	3D	RC	Inverted Pendulum	2010	UC San Diego	Various	3	200
Squat RC Shear Wall at ELSA	Seismic	2D	RC	Shear Wall Comp.	2012	ELSA, JRC Ispra	OpenSees	1	0







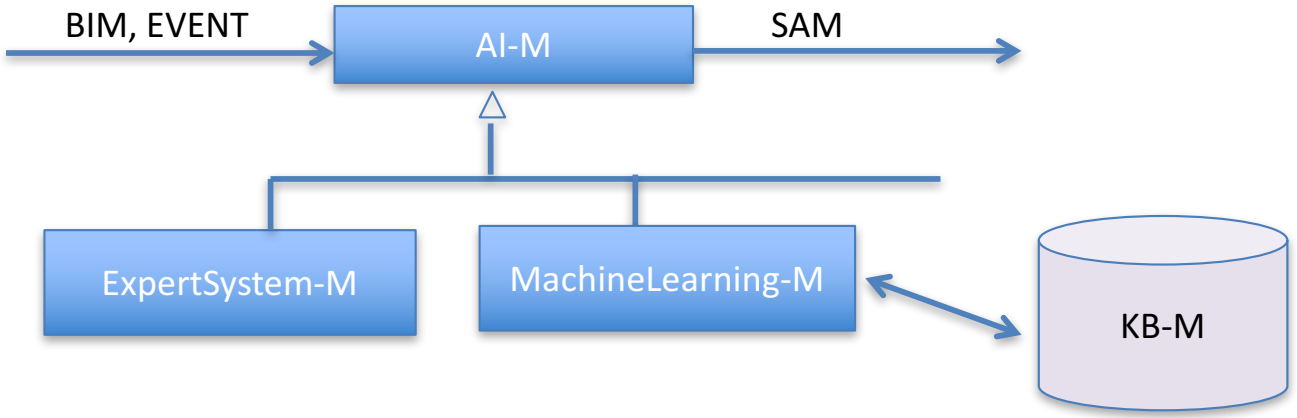


& Converted to SimCenter JSON File Format

WE NEED DATA

AI-M-1: Machine Learning application that will use related Experimental and Simulation Data to Generate SAM files

Stella Yu



Challenge: Is to turn BIM and SAM files into a numerical representation on which machine learning algorithms can be turned loose.

Current Working Approach: based on component level approach to modeling:

1. From paired (BIM,SAM) files identify and model corresponding components.
2. Knowing how each component is modeled, create a model representation given BIM
3. Generate possible SAM models for each component in the BIM

Research Tools Release Schedule

uqFEM

- V1.0 (June 2018)
- V2.0 (2019)
- V3.0 (2020)
- V4.0 (2021)

CWE-UQ

- V1.0 (June 2018) Bluff Body
- V2.0 (Sept 2018) Building
- V3.0 (2019) UQ

AI

EE-UQ

- V1.0 (July 2018) Uniform
- V2.0 (Sept 2018) Rock Outcrop
- V3.0 (2019) Soil Box

AI

PBE

- V1.0 (Sept 2018) Earthquake
- V2.0 (2019) Wind
- V3.0 (2020) Water

AI

Workflow Testbeds Release Schedule

RDT

- V1.0 (2019) Earthquake
- V2.0 (2020) Wind
- V3.0 (2021) Water

Regional Earthquake

- V1.0 (June 2018)
- V2.0 (2019)
- V3.0 (2020)

Water

- V1.0 (Sept 2018)
- V2.0 (2020)
- V3.0 (2021)

Hurricane

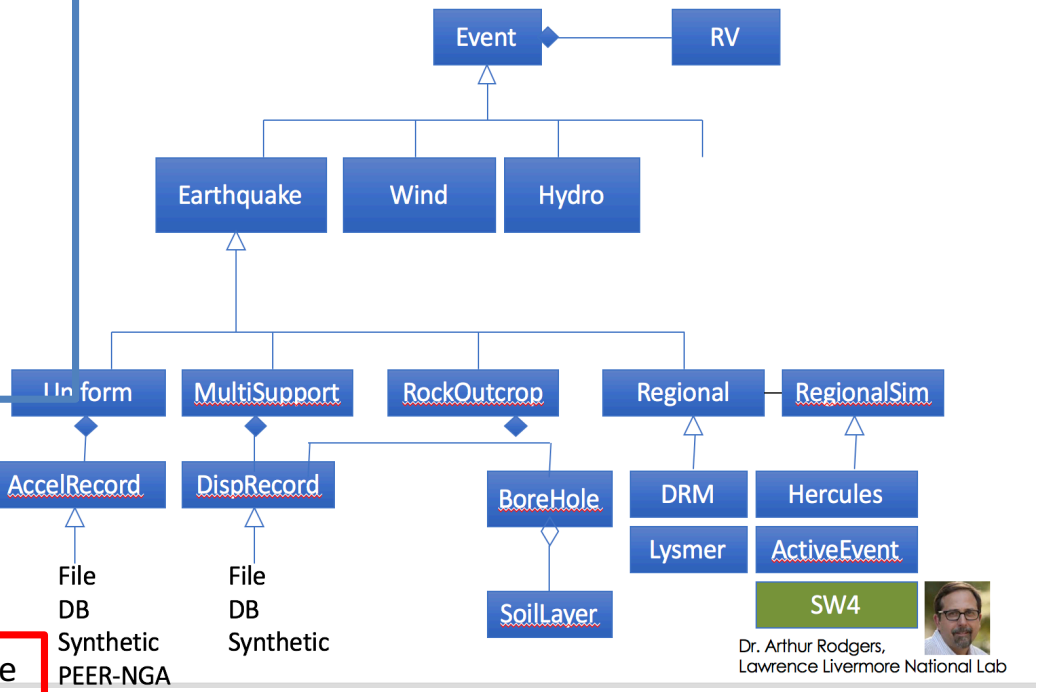
- V1.0 (2020)
- V2.0 (2021)

MultiHazard

- V1.0 (2021)

Tool to Predict the Response of a Building to an Earthquake Event

- Moment Frames, Braced Frames, and Concrete Shear Wall Buildings
- User to specify random variables and to **obtain measure of uncertainty in computed responses**
- To include **SSI**
- Jobs to run locally or **remotely at DesignSafe**.

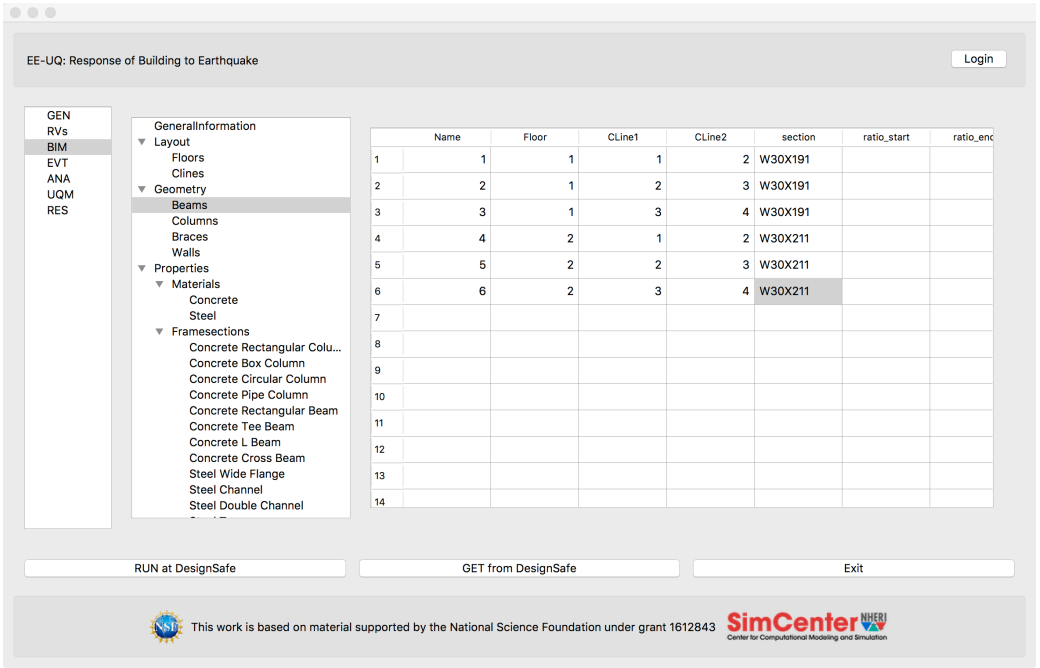


Release Dates:

- V1.0 (June 2018) Uniform Excitation + Moment Frame
- V2.0 (Sept 2018) Rock Outcrop motion
- V3.0 (?????????) Soil Box around Building

PBE Tool to Predict the Damage & Loss to a Building given some Hazard Event

- User Inputs Building Information
  - Structural & Contents
- User Selects from different loading options & Inputs Parameters
- User Specifies RV distributions
- The tool when run will auto generate the analysis model, run a set of deterministic simulations and use Pact or equivalent to generate D&L all on DesignSafe
- User selects run & views different output results.



Release Dates:

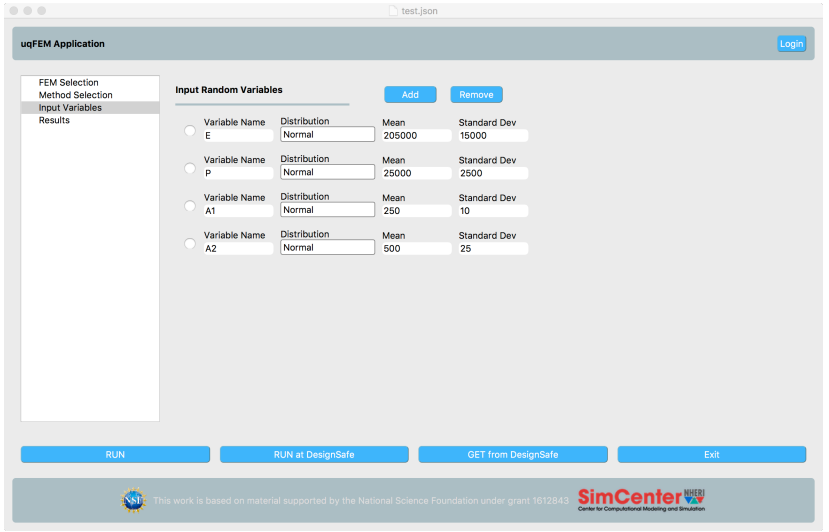
- V1.0 (Sept 2018) Earthquake
- V2.0 (2020) Wind
- V3.0 (2021) Water

Research Opportunities:

- In addition to modeling & hazards, different applications for calculating damage & loss
- Validated data for different fragility and loss curves.

Tool to add UQ & Optimization Capabilities to Existing FE code.

- User provide input file & script for an existing FE application
- Application reads & creates random variables for parameters specified in input
- User defines the UQ or Optimization Method (Sampling, Parameter Estimation, Bayesian Calibration)
- User specifies the distributions on the Random Variables
- User submits to local machine or remotely to DesignSafe-ci



Release Dates:

- **V1.0 (June 2018). .. IT's AVAILABLE NOW!!!!!**
- V2.0 (2019)
- V3.0 (2020)

Takes as Input Standard FE input file

exampleOpenSees — emacs TrussTemplate.tcl — 90×45

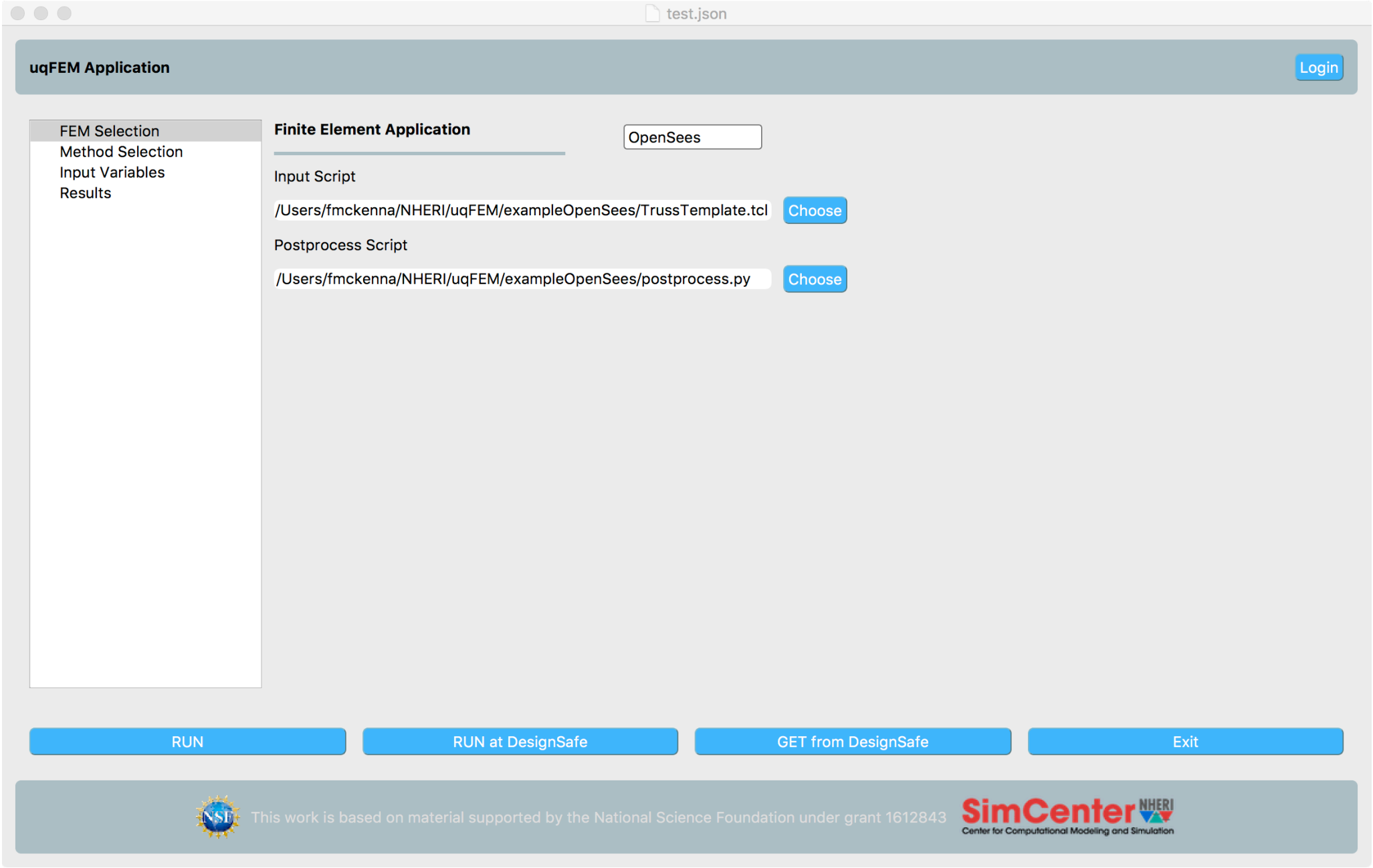
```
FEAP * * Elastic Truss
6 9 2 2 2 2

PARAmeter
E       = 205000
P       = 25000
A2      = 500
A1      = 250

MATErial 1
TRUSs
ELAStic isotropic E
CROSs section A1

MATErial 2
TRUSs
ELAStic isotropic E
CROSs section A2

COORdinates
1       0       0.0             0.0
2       0       4e+3            0.0
3       0       8e+3            0.0
4       0       12e+3           0.0
5       0       8e+3            4e+3
6       0       4e+3            4e+3

ELEMents
1       0       1       1       2
2       0       1       2       3
3       0       1       3       4
4       0       2       4       5
5       0       2       5       6
6       0       2       6       1
7       0       1       2       6
8       0       1       6       3
9       0       1       3       5

BOUNdary
1       0       1       1
4       0       0       1

FORCe
-uu-(DOS)---F1  TrussTemplate.txt    Top L1      (Te
```

exampleOpenSees — emacs TrussTemplate.tcl — 90×45

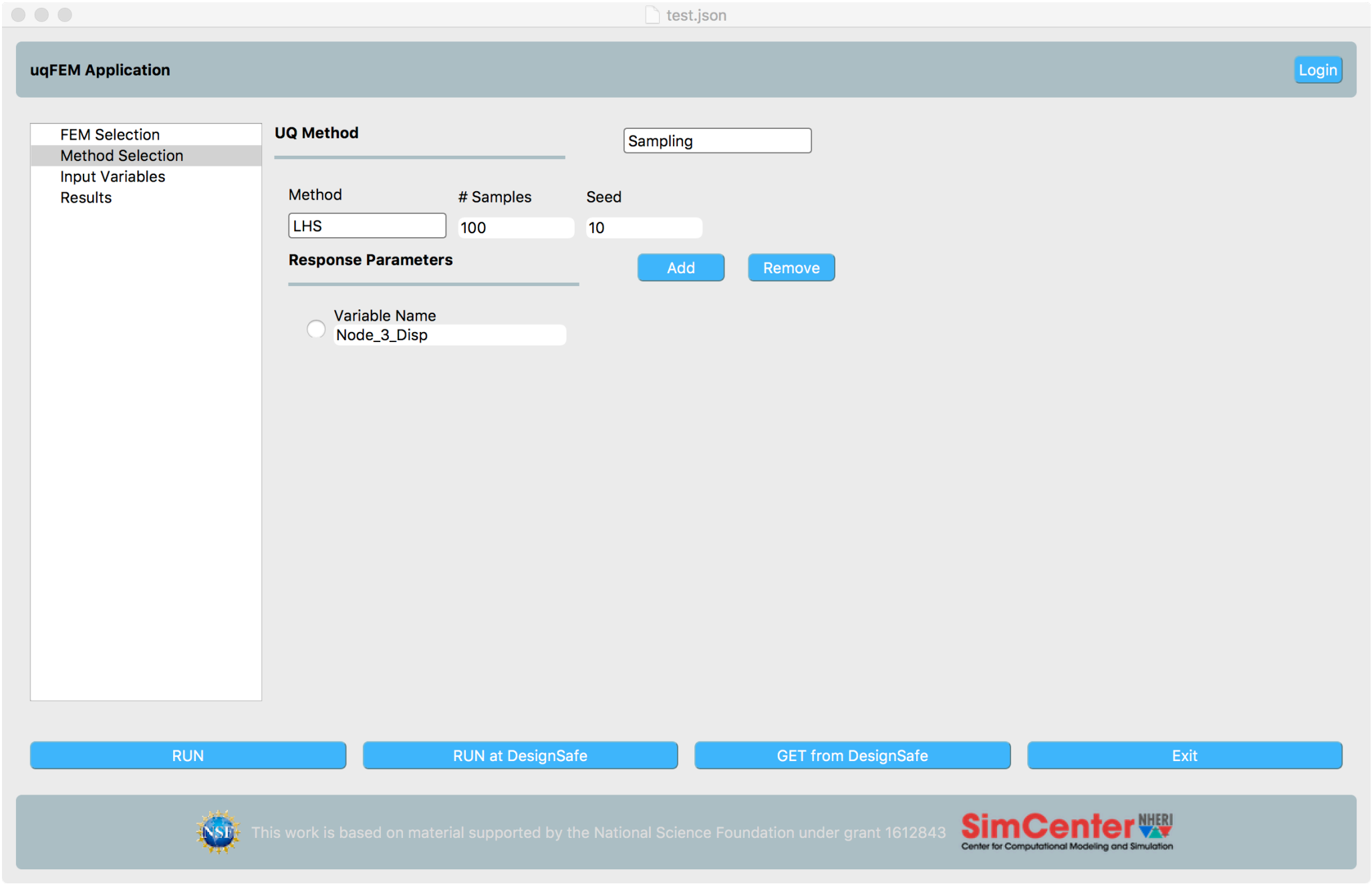
```
pset E 20500
pset P 25000
pset A1 250
pset A2 500

model Basic -ndm 2 -ndf 2
node 1     0    0
node 2  4000    0
node 3  8000    0
node 4 12000    0
node 5  4000 4000
node 6  8000 4000
fix 1 1 1
fix 4 0 1
uniaxialMaterial Elastic 1 $E
element truss 1 1 2 $A1 1
element truss 2 2 3 $A1 1
element truss 3 3 4 $A1 1
element truss 4 1 5 $A2 1
element truss 5 5 6 $A2 1
element truss 6 6 4 $A2 1
element truss 7 2 5 $A1 1
element truss 8 3 6 $A1 1
element truss 9 5 3 $A1 1
timeSeries Constant 1
pattern Plain 1 1 {
    load 2 0 -$P
    load 3 0 -$P
}

recorder Node -file node.out -node 1 2 3 4 5 6 -dof 2 disp

algorithm Linear
integrator LoadControl 1.0
system ProfileSPD
numberer RCM
constraints Plain
analysis Static
analyze 1
```

```
-uu-(DOS)---F1  TrussTemplate.tcl    All L1      (Tcl)----------------------------------
```

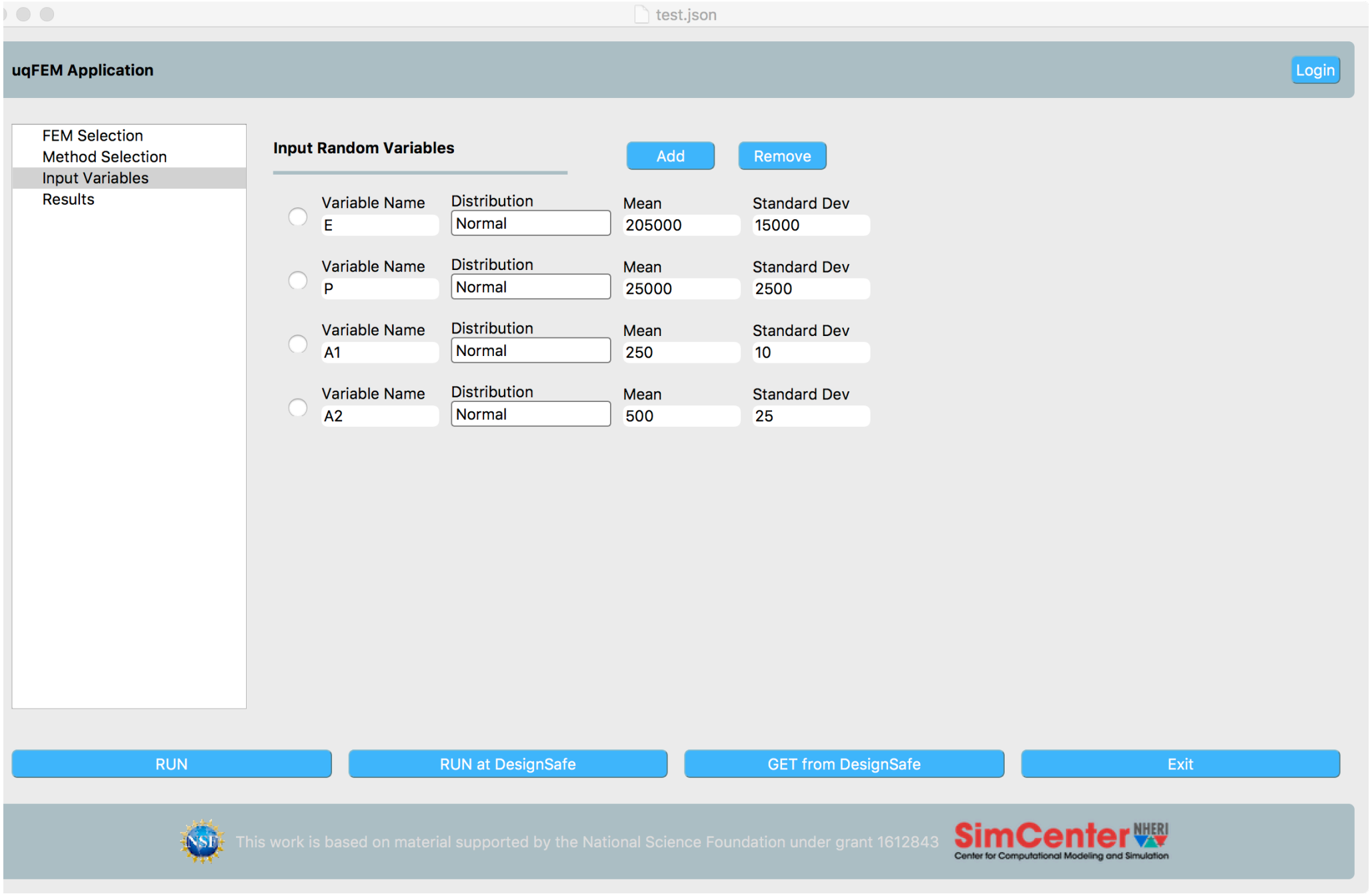
User Selects FE Application and Input File



User Specifies UQ Method & EDP



User Edits Random Variables



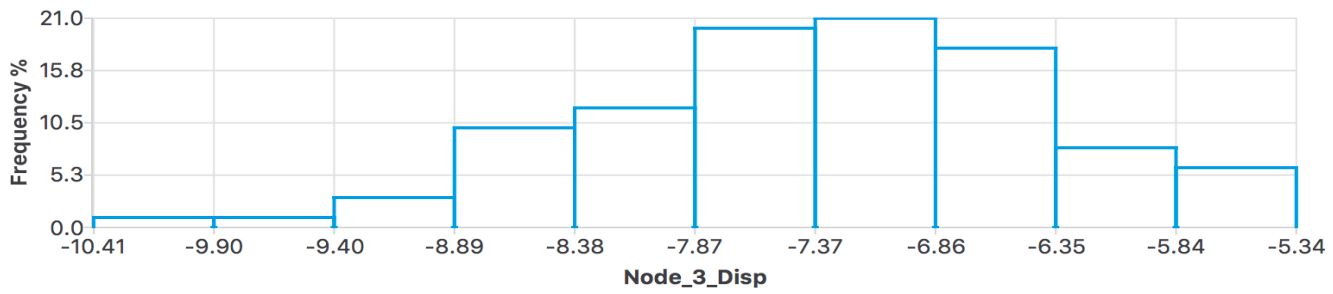
User Runs & Looks at Results

test.json

uqFEM Application Login

- FEM Selection
- Method Selection
- Input Variables
- Results

Summmary | General | Data Values



	Run #	E	P	A1	A2	Node_3_Disp
1	1	206909.7696	21737.61704	254.9364721	455.1905439	-6.52346
2	2	197575.2597	25299.20459	260.3806336	488.5760863	-7.59824
3	3	204934.6077	19431.52982	254.0986264	519.5920637	-5.53646
4	4	222499.436	25223.16776	239.0705048	524.2251988	-6.81669
5	5	221794.7483	23205.00702	269.0889998	510.3215642	-5.97665
6	6	212996.0309	24896.34298	242.6211375	482.3671031	-7.24182
7	7	185415.0218	23905.72273	234.8309478	508.893707	-7.9343
8	8	229505.5419	25531.46523	253.6452697	470.5777446	-6.81369

RUN | RUN at DesignSafe | GET from DesignSafe | Exit

This work is based on material supported by the National Science Foundation under grant 1612843

SimCenter NHERI

Exercise

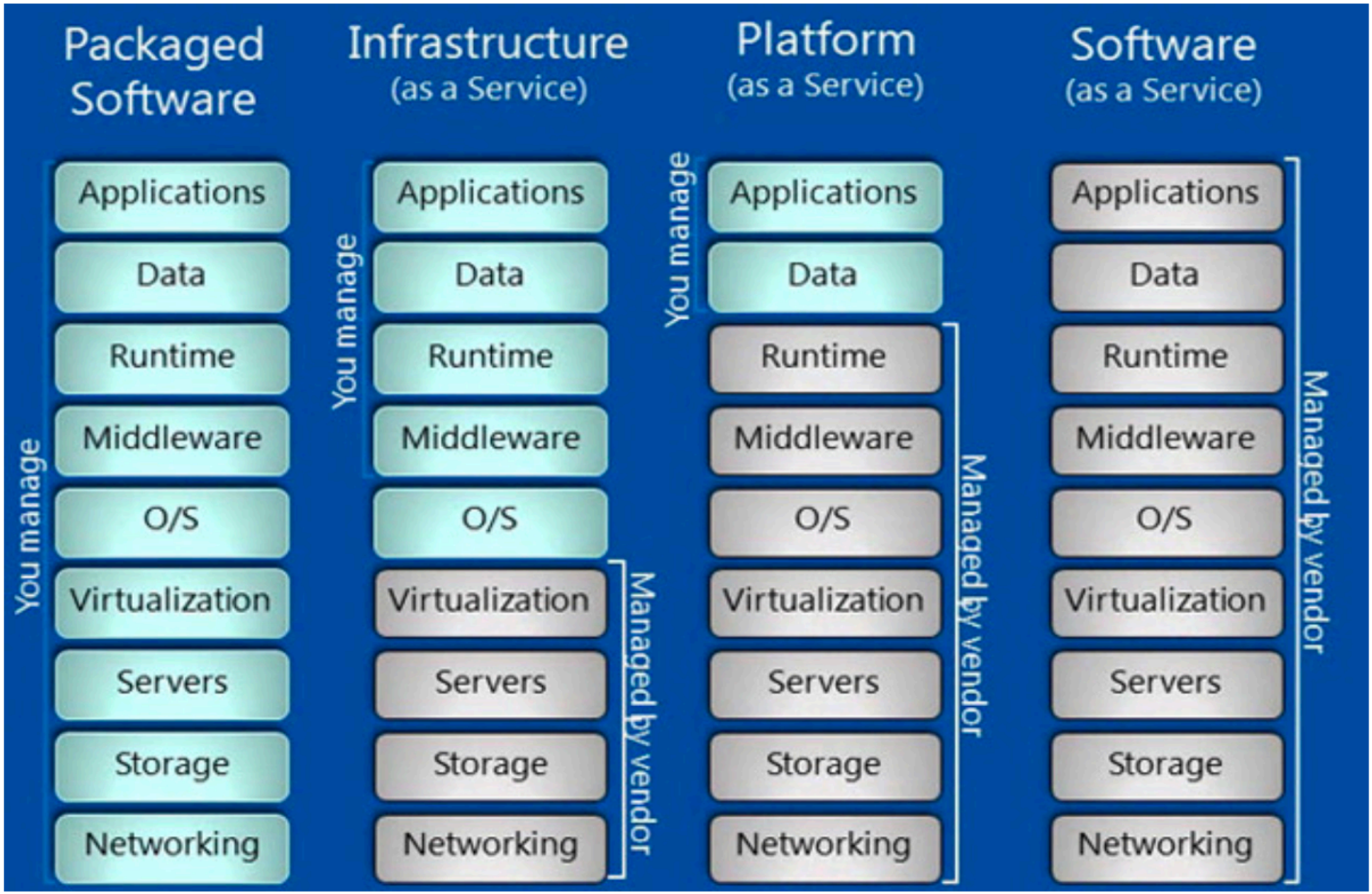
TEST uqFEM

Cloud Computing

Cloud Computing is using the internet to access applications, data or services that are stored or hosted on remote services.

Anybody offering such services over the internet is a cloud provider.

Cloud Providers Are Classified in 1 of 3 ways depending on what they provide.



What is Agave?

Agave is a *Science-as-a-Service* cloud API platform

- **Run scientific codes**

  your own or community provided codes
- **...on HPC, HTC, or cloud resources**

  your own, shared, or commercial systems
- **...and manage your data**

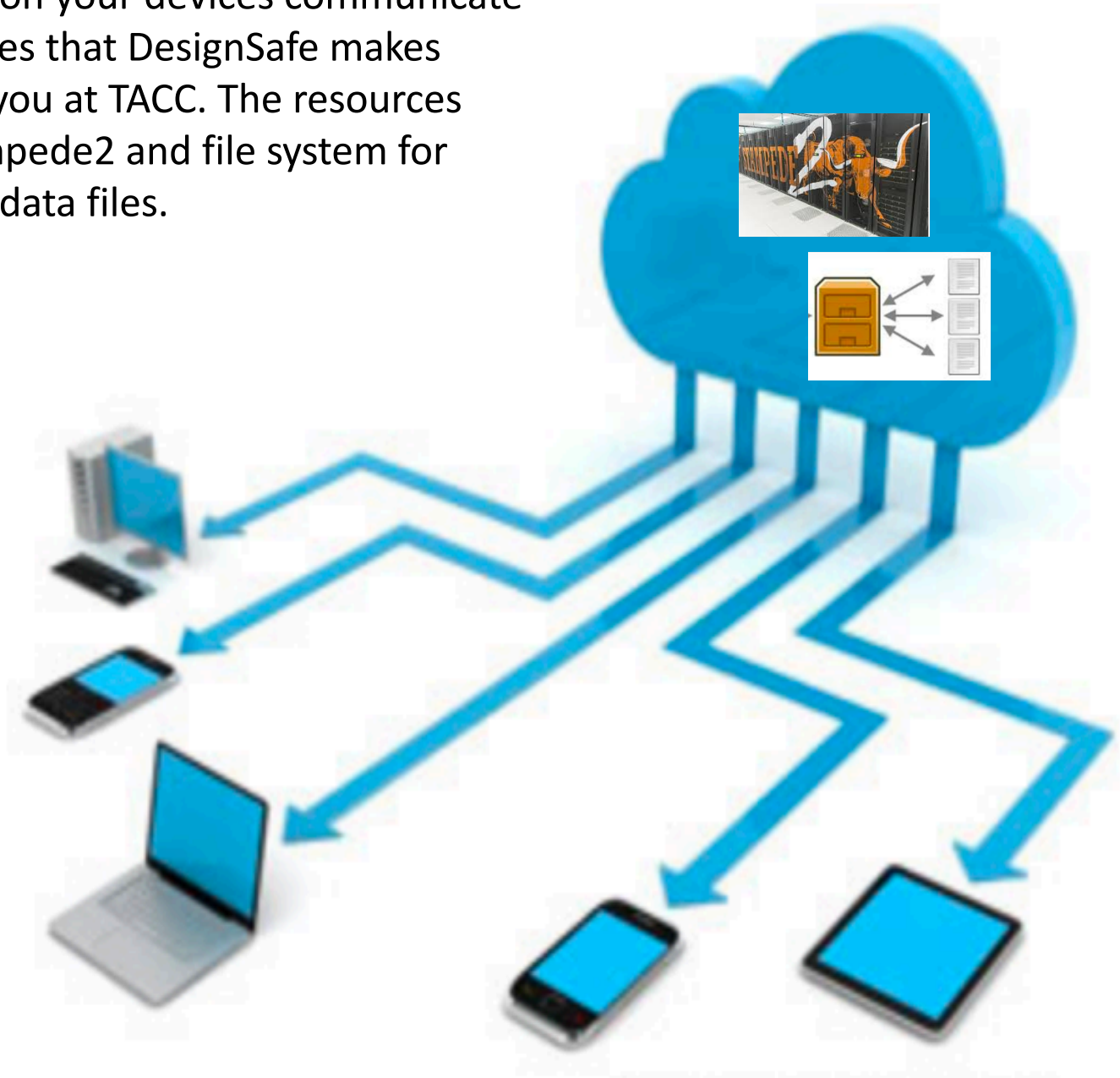
  reliable, multi-protocol, async data movement
- **...from the web**

  webhooks, rest, json, cors, oauth2
- **...and remember how you did it**

  deep provenance, history, and reproducibility built in

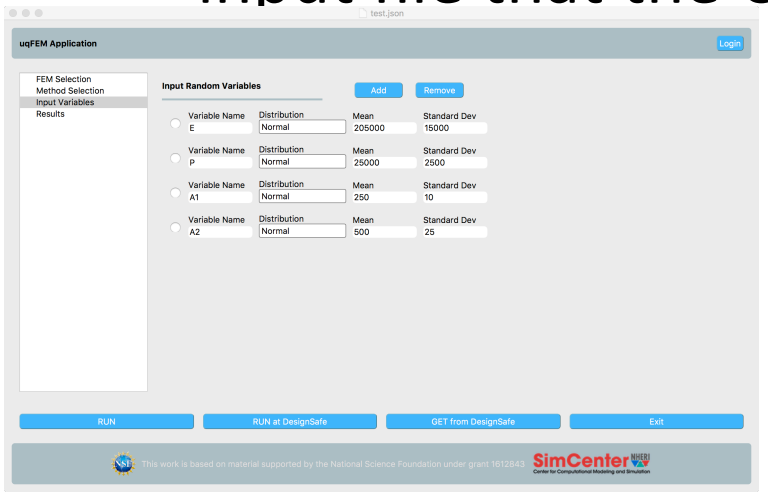
TACC

Agave is the interface through which the applications on your devices communicate with resources that DesignSafe makes available to you at TACC. The resources include Stampede2 and file system for storing your data files.



SimCenter Apps X 2

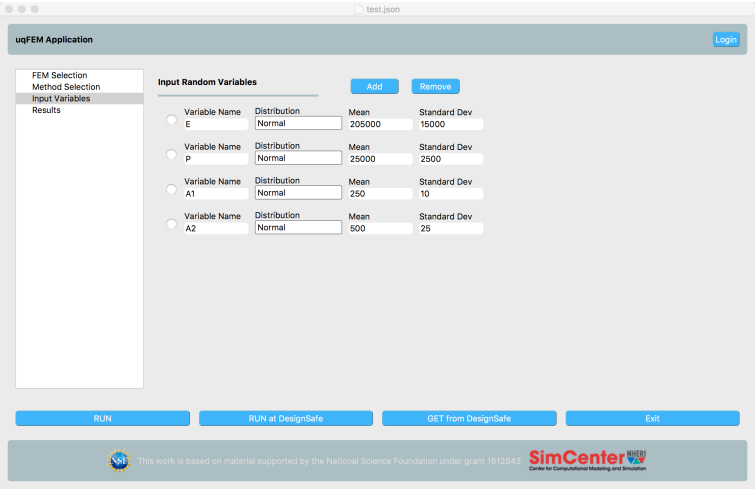
- We are actually developing 2 sets of apps
  1. UI Frontend built with Qt
  2. Backend python and “Pegasus” workflows to run the input file that the UI produces.



Click the “Run” and operation:

1) Outputs dakota.out JSON file outlining entries
2) Invokes a Python script in application dir
3) This script creates a dir under your, copies files and makes dakota input
4) Script then invoked Dakota
5) When run, control back to app which loads output files

SimCenter Apps X 2



Click the “Run at Design-Safe” and operation:

1) Outputs dakota.out JSON file outlining entries
2) Invokes a Python script in application dir
3) This script creates a dir under your, copies files and makes dakota input
4) Control returns to app,
5) IN A THREAD, the App then starts sending the files to your DesignSafe dir
6) Once files are there, the thread creates a signal to bring up popup for getting job data

There are C++ classes we developed to interface with Agave from C++ code

AgaveCLI	AgaveCurl	RemoteDataInterface
Uses the cli sequential	Uses libCurl **threaded**	Uses https **threaded**

Basic functionality the same:

- sendFiles
- startApplication
- checkJobStatus
- getFiles

Core APIs

/systems /files /apps /jobs /meta

- Register storage and compute systems
- Ingest, move and transform data files and folders
- Register applications (binaries) on large systems
- Launch jobs to invoke applications
- Capture metadata about the experiment

Agave Commands

- What follows are Agave cli (command line) commands
- The list of commands will be in agave-cli/bin directory
- All commands relevant to a particular part of agave start with same name, e.g. files-, system-, jobs-, apps-,

```
command -h
```

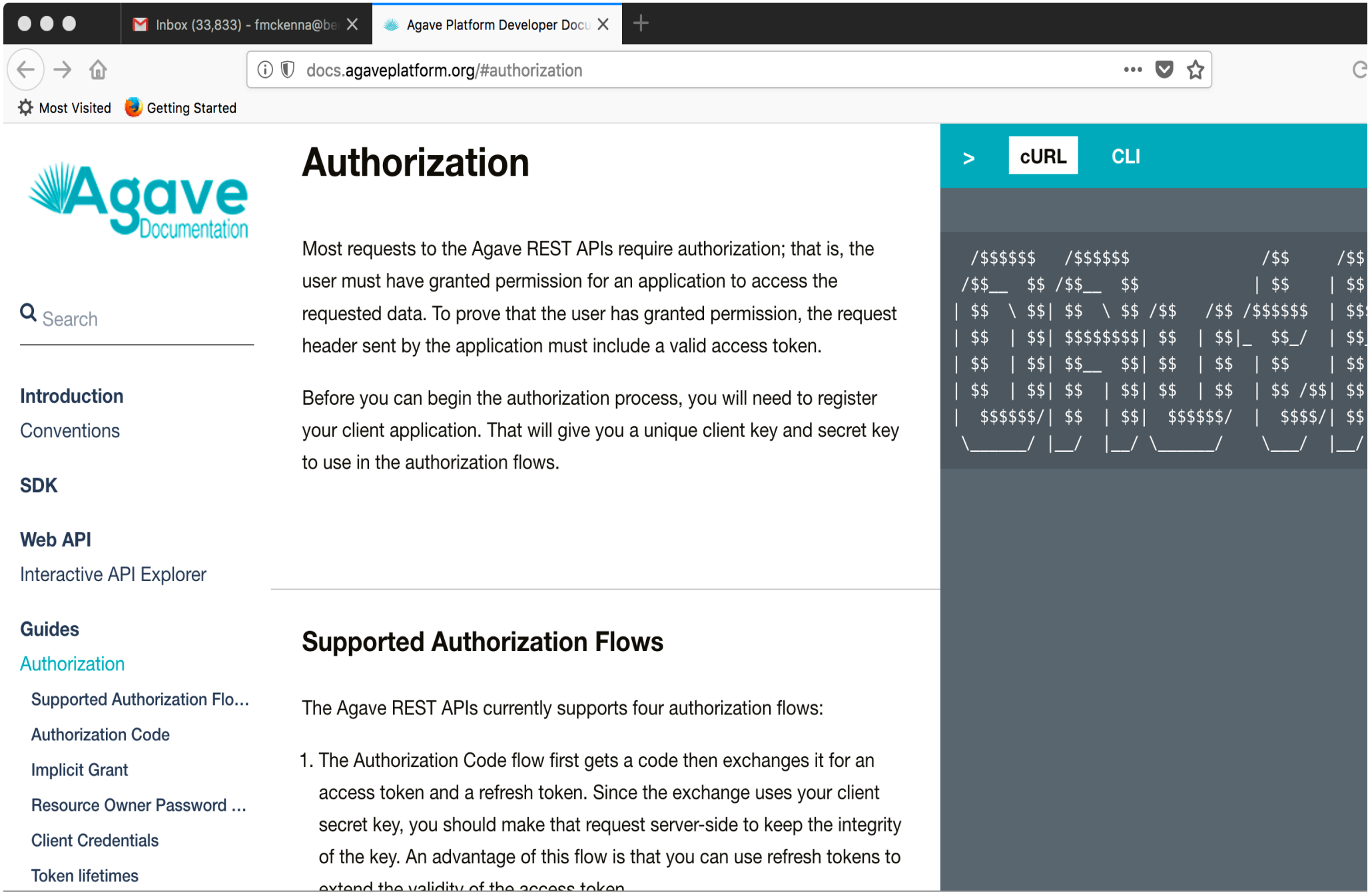
- The commands all produce output .. verbose mode

```
command -v
```

- The commands all produce output .. VERY verbose mode

```
command -V
```

Documentation is Great (Examples are Sparse)



Agave Set Up Exercise

- There are 4 packages you need to install on the VM (password:ws2018)

```
sudo apt. install git
sudo apt install curl
sudo apt install python
sudo apt istall python-3
```

- Need to now edit file and clone a repository

```
gedit .bashrc
At end of file .bashrc place line:
                    PATH=$PATH:/home/simcenter/agave-cli/bin

git clone https://bitbucket.org/agaveapi/cli.git   agave-cli
```

- Lastly 2 Agave commands to get you authorization

```
tenants-init -t designsafe
clients-create -S -v -N myClient -D "myClient"
```

Get a token for access to Designsafe through Agave for awhile

```
auth-tokens-create -S –v
```

```
auth-tokens-refresh -S –v
```

/systems /files /apps /jobs

- Systems – a server or cloud of servers.
- Two types of Systems
  1. Storage - for storing and manipulating files
  2. Execution systems – for running applications

```
systems-list
systems-list –S (list storage systems)
systems-list –E (list execution systems)
systems-list -v designsafe.storage.community
```

```
auth-tokens-refresh -S –v
```

/systems /files /apps /jobs

Files Operations on your default storage system

```
files-list  fmk
files-mkdir  -N dir1   fmk
files-mkdir –N dir2   fmk/dir1
files-upload -F  OpenSeesExampleMP.tar.gz    fmk/dir1
files-upload -F  drivel    fmk/dir1/dir2
files-list   fmk   dir1/dir2
files-get –N moreDrivel   fmk/dir1/dir2/drivel
files-delete  fmk/dir1/dir2
```

Of course replace fmk with yourname

/systems /files /apps /jobs

List the Apps Available to You

```
apps-list
apps-list –v opensees-MP-2.5.0.6606u9
apps-list –v opensees-MP-2.5.0.6606u9 > tmp
```

/systems /files /apps /jobs

```
jobs-list
jobs-list -v 4416860733070306840-242ac11b-0001-007
```


```
{
  "id": "4416860733070306840-242ac11b-0001-007",
  "name": "OpenSeesMP demo",
  "owner": "tg457427",
  "appId": "pi-3.0.0.6709",
  "executionSystem": "designsafe.tg457427.exec.stampede2",
  "batchQueue": "normal",
  "nodeCount": 1,
  "processorsPerNode": 32,
  "localId": "1860269",
  "created": "2018-08-03T03:46:55.000-05:00",
  "lastUpdated": "2018-08-03T03:55:57.000-05:00",
  "archivePath": "tg457427/archive/jobs/job-4416860733070306840-242ac11b-0001-007",
```

Submit Some Jobs

Code/Cloud/Agave/jobs/submitExtract.json

```
{
    "name":"extract directory",
    "appId":"extract-0.1u1",
    "parameters":{
        "inputFile":"fmk/dir1/OpenSeesExampleMP.tar.gz"
    },
    "notifications": [
            {"url" : "fmckenna@berkeley.edu",
            "event": "*"
            }
    ]
}
```

```
jobs-submit –v  –F  submitExtraction.json  >  tmp1
```

```
jobs-status –v  1265736004352217576-242ac11b-0001-007
```

Code/Cloud/Agave/jobs/submitOpenSees.json

```
{
    "name":"extract directory",
    "appId":"extract-0.1u1",
    "parameters":{
        "inputFile":"fmk/dir1/OpenSeesExampleMP.tar.gz"
    },
    "notifications": [
            {"url" : "fmckenna@berkeley.edu",
            "event": "*"
            }
    ]
}
```

```
jobs-submit –v  –F  submitOpenSees.json  >  tmp1
```

```
jobs-status –v  1265736004352217576-242ac11b-0001-007
```

If Interested in setting up own app

1. Setting Up SSH keys

- On stampede:

```
ssh-keygen (don't provide password)
cd .ssh
cp id_rsa.pub authorized_keys
```

- On your machine:

```
cd ~
mkdir .ssh
scp your_login@stampede2.tacc.utexas.edu:~/.ssh/id* ./
```

- Test it, on your machine still

```
ssh login@stampede2.tacc.utexas.ede
```

2. Set up exe service

```
systems-clone –v -I designsafe.fmk.stampede2.
    designsafe.community.exec.stampede2.nores
```

systems-list –v

Edit designsafe.fmk.stampede2 I Provide

Your username, HOME and SCRATCH directories and from tmp copy the SSH key stuff

systems-addupdate –v –F designsafe.fmk.stampede2

3. Clone an App

- apps-clone -e **designsafe.tg457427.exec.stampede2** -n pi OpenseesMp-3.0.0.6709u2

```
jobs >apps-clone -e designsafe.tg457427.exec.stampede2 -n pi OpenseesMp-3.0.0.6709u2
Successfully cloned app OpenseesMp-3.0.0.6709u2 to pi-3.0.0.6709
```

the cloned app should point to your exe system

- copy submitOpenSees.json to submitPI.jsonChange the id of the service in submitPI.json
- jobs-submit –F submitPI.json

4. Create the pi exe

On stampede pull on your git repo

- git pull

Compile the file Code/Cloud/Agave/mpi

- mpicc pi.c –o pi

5. Create the PI Service

- apps-list -V pi-3.0.0.6709

Shows deployment path

"deploymentPath" : ”fmk/applications/pi-3.0.0.6709",

files-list fmk/applications/pi-3.0.0.6709

- files-get –r -n wrapper.sh tg457427/applications/pipi-3.0.0.6709/wrapper.sh

Change OPENSEES_BIN line to point to pi exe

- files-put –F wrapper.sh -n wrapper.sh tg457427/applications/pi-2.5.0.6606

Change validator for input script in json file to “”

TEST IT