

A complex network diagram with numerous nodes and connecting lines, rendered in a light gray color against a dark gray background. The nodes are small circles, and the lines are thin, creating a dense web of connections that fills the entire frame.

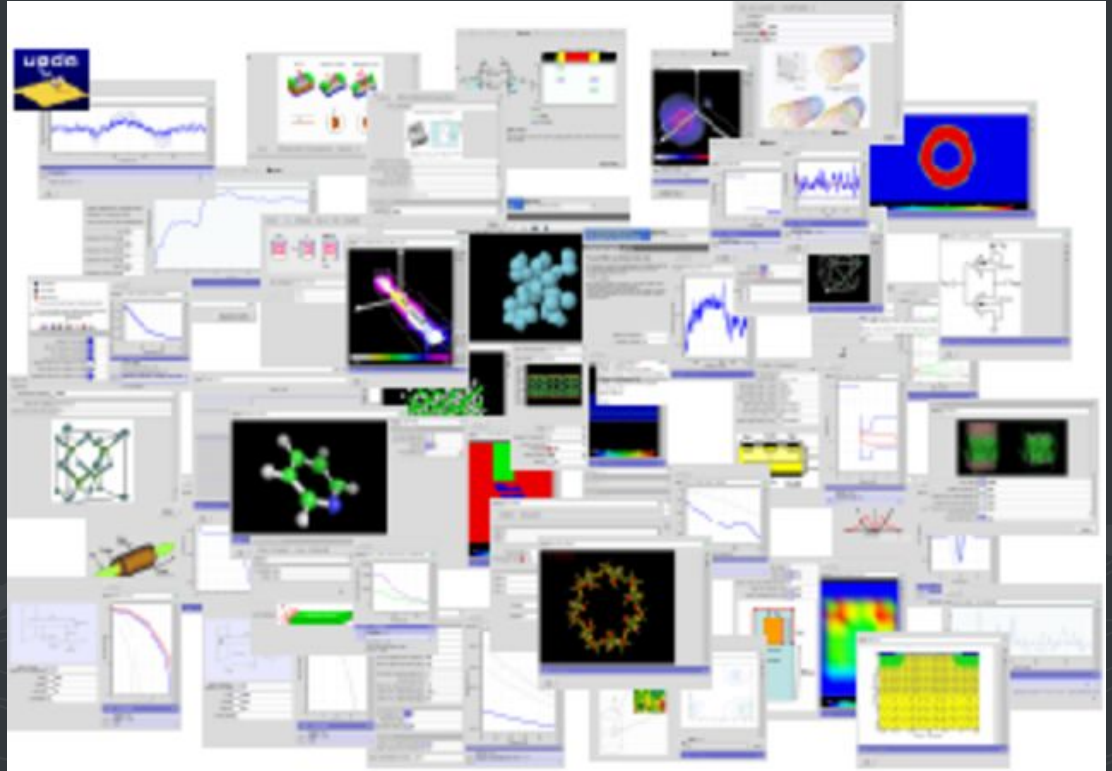
(Re)Enabling BOINC as a nanoHUB Computational Platform

Steven Clark
nanoHUB.org
HUBzero.org

Research Computing, Purdue University



- Collaborate
- Simulate
- Explore
- Learn



nanoHUB SIMULATION USE CASES

- **On demand**

- UI used to declare inputs for simulation
- Command line
- GPU

- **Cache resolution**

- Input (driver.xml) files are placed in a cache backlog queue
- External process
 - pulls input from cache backlog queue
 - does the simulation
 - saves the result
- If cache result exists no simulation is required simply pull the existing result
- Faster response time provides better user experience

nanoHUB SIMULATION USE CASES

- **Parametric Sweeps**

- `submit -p @@vth=0:0.2:5 -p @@cap=10pf,100pf,1uf sim.exe @:indeck`

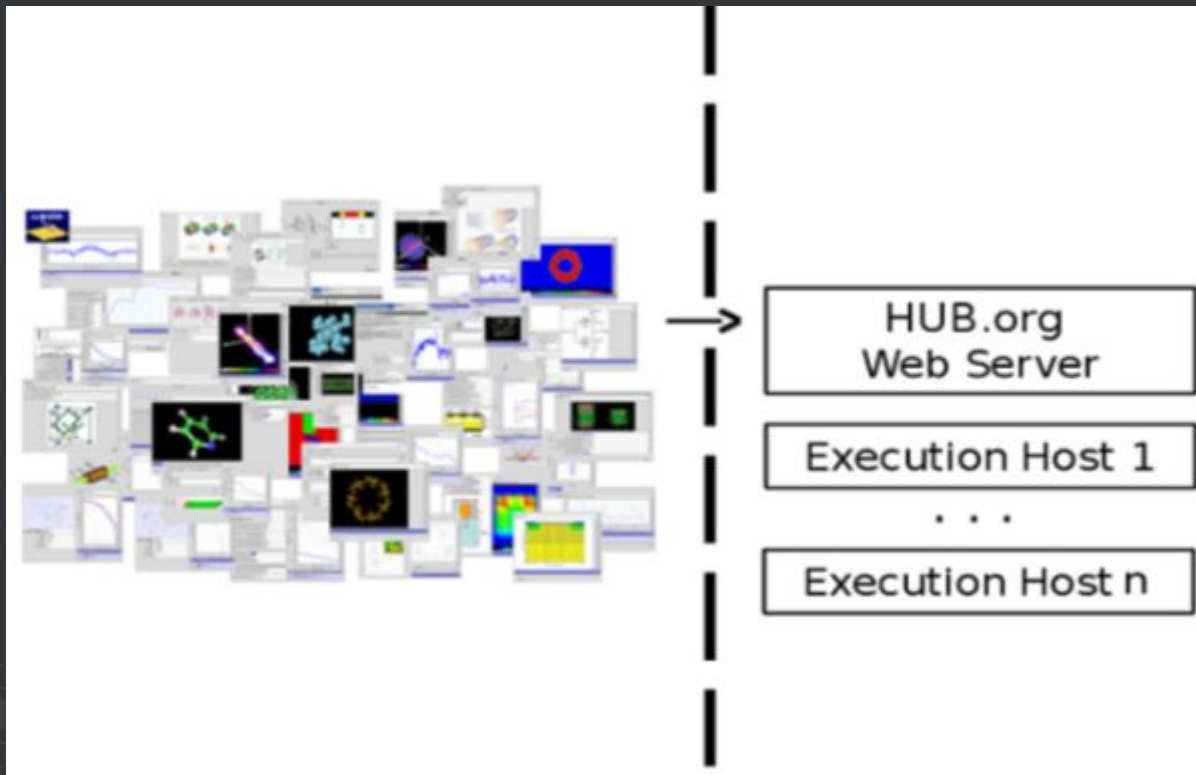
- **Uncertainty quantification (UQ)**

- Inputs declared as distributions
- Statistical methods used to determine input samples
- `submit --data input.csv sim.exe @:indeck`
- A simulation is run for each sample
- Result is a response surface model which can be used to approximate simulation at greatly reduced cost

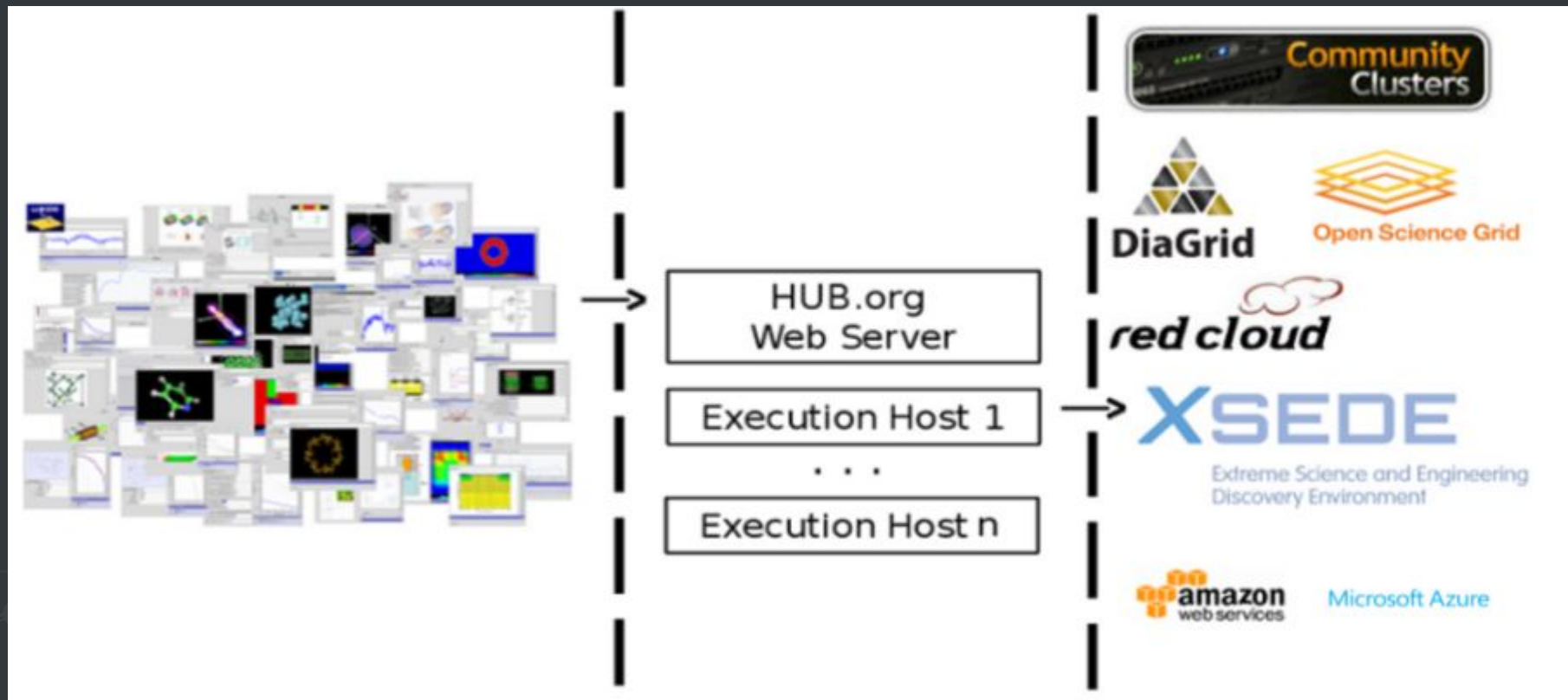
- **Exploratory simulation**

- Explore multidimensional input space
- Generate simulation input samples covering the space
- `submit --data input.csv sim.exe @:indeck`
- Execute simulation for each sample

SUBMITTING JOBS TO LOCAL RESOURCES



SUBMITTING JOBS TO HPC/HTC RESOURCES



SUBMIT INFRASTRUCTURE

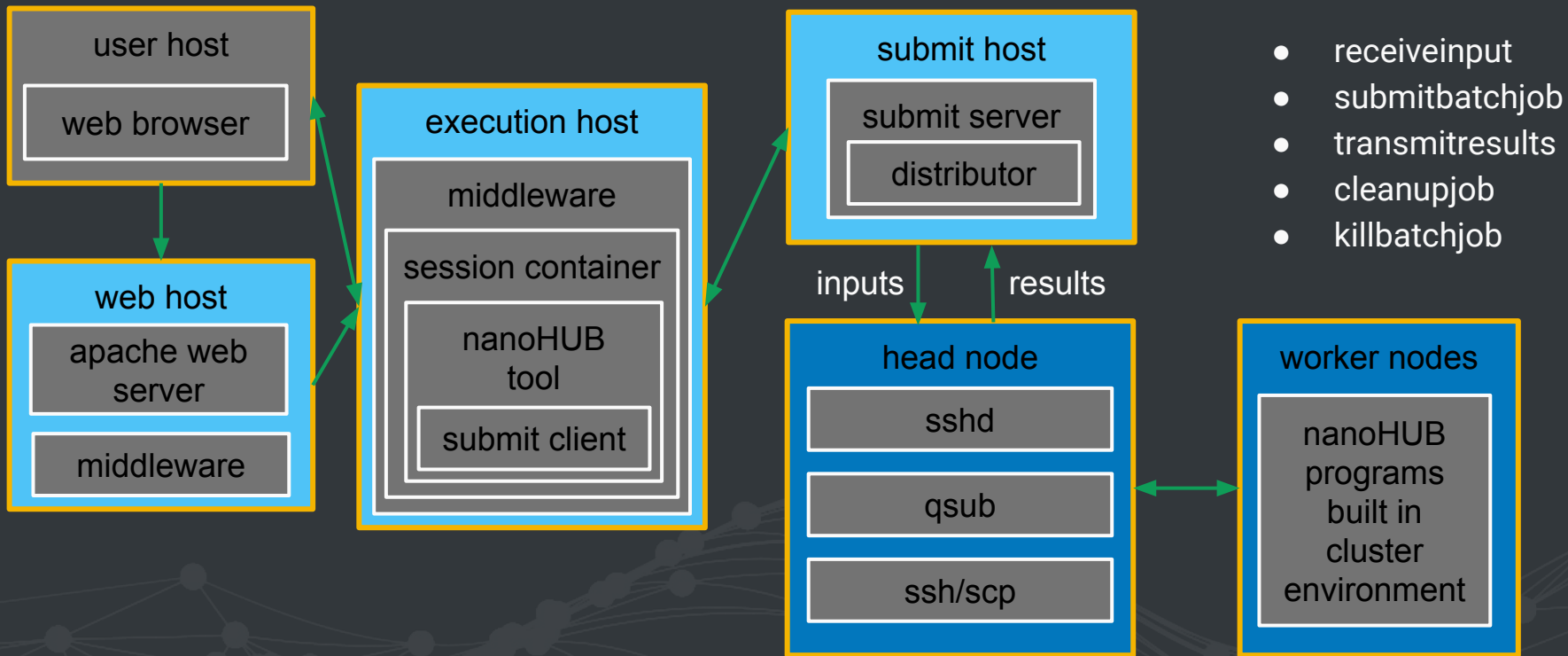
- Proxied job submission with community account
- Standard file transport via ssh/tar



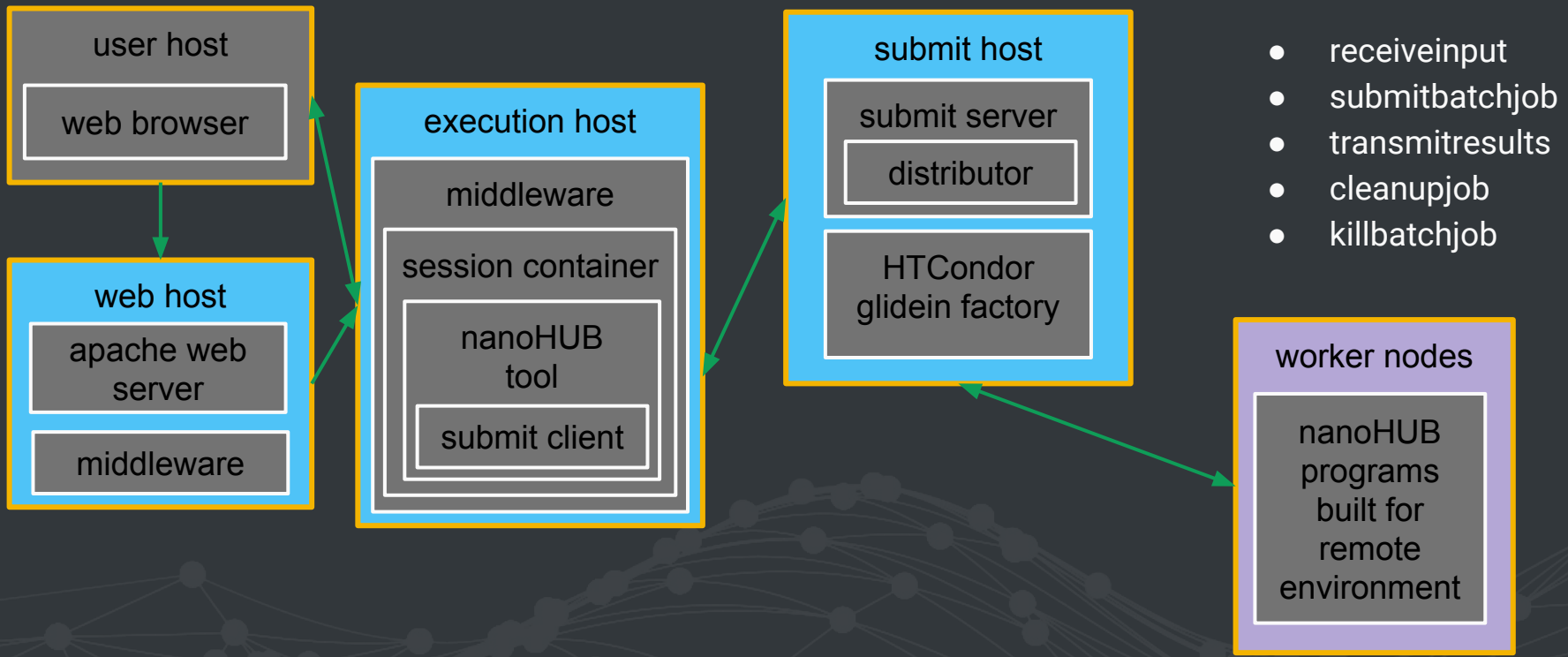
- Interface with several batch schedulers
- Configurable
 - Sites
 - Tools
 - Identity
 - Application Access



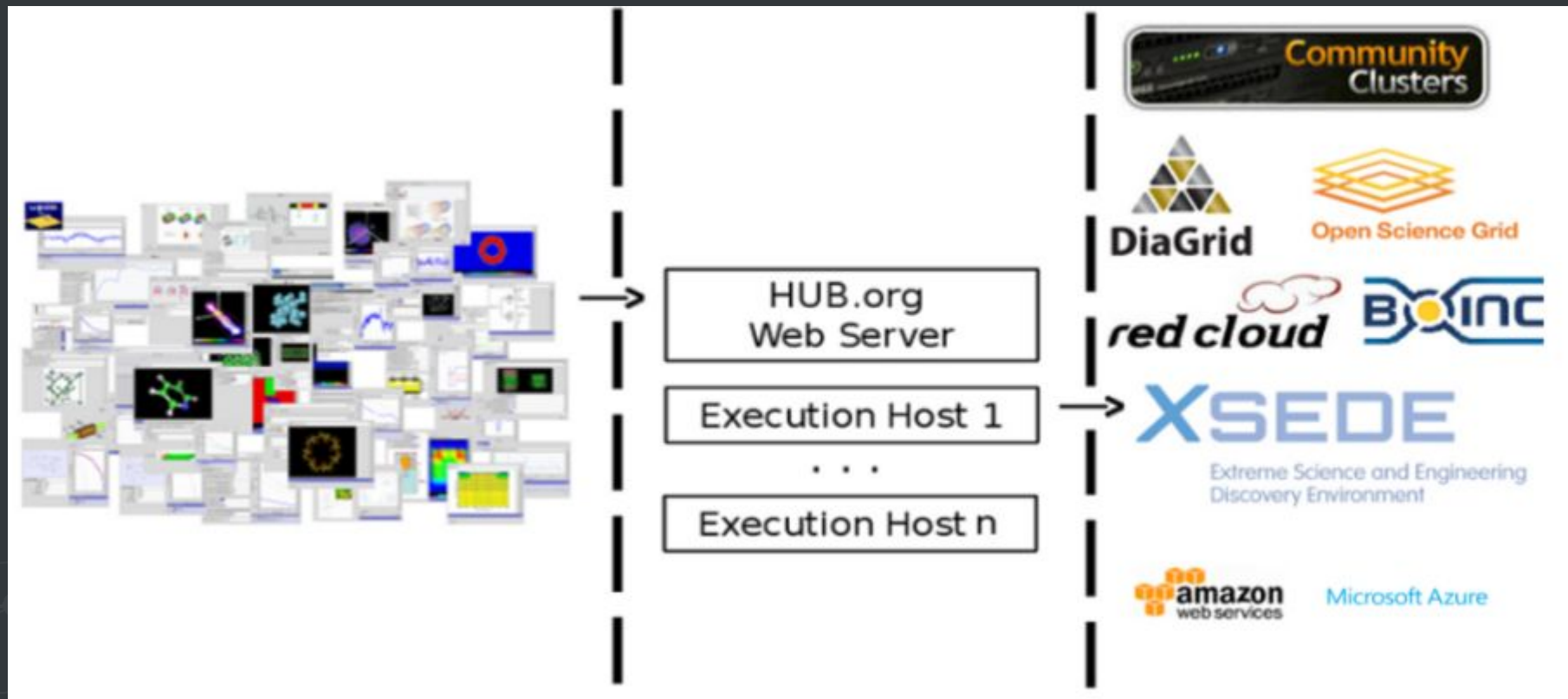
ARCHITECTURE - nanoHUB/HPC



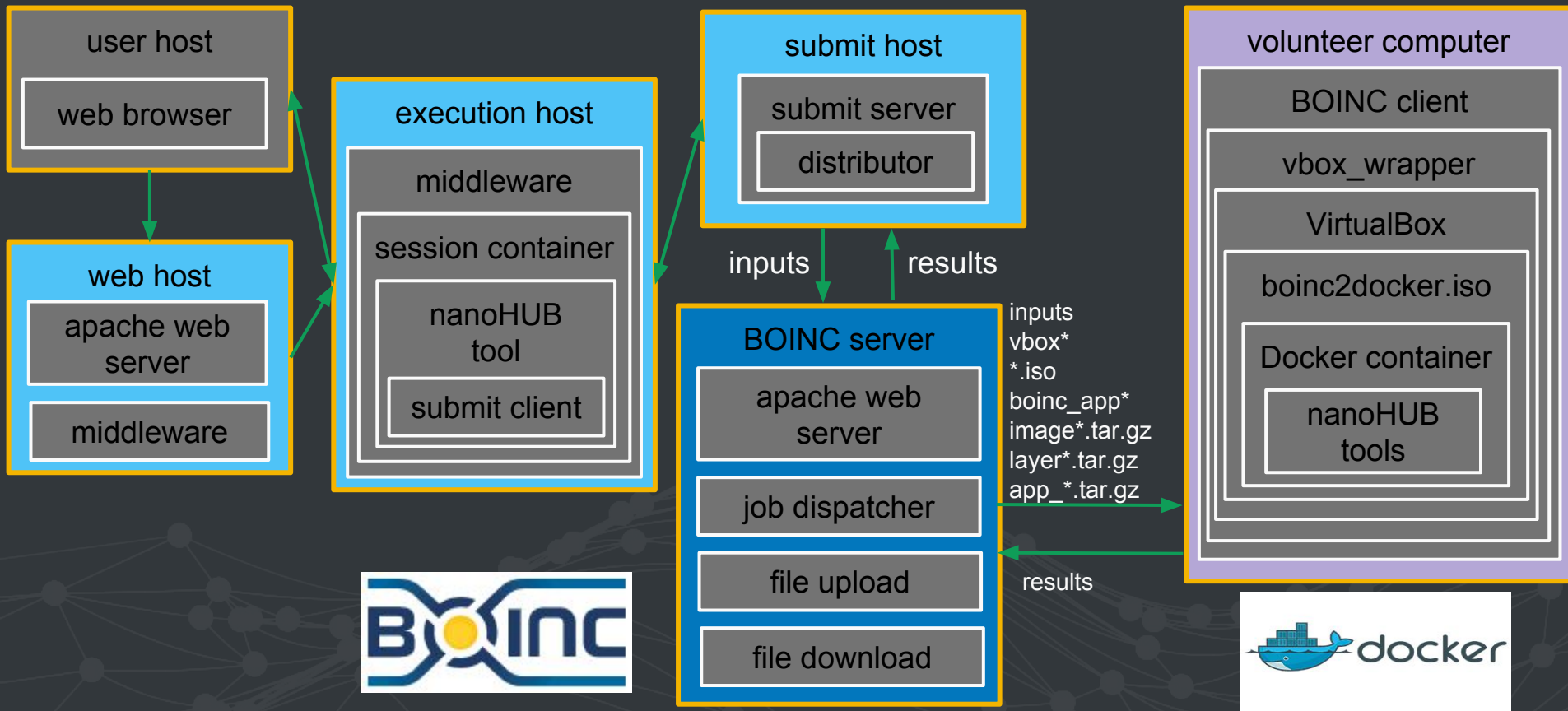
ARCHITECTURE - nanoHUB/HTC



SUBMITTING JOBS TO HPC/HTC RESOURCES



ARCHITECTURE - nanoHUB/BOINC



BOINC - APPROACH

- **Volunteer Host** - where the work happens
 - VirtualBox - nanoHUB applications run in Linux environment. VirtualBox provides access to Windows and MAC volunteer hosts.
 - boinc2docker - introduction of docker containers allows simpler change management. One docker container can support many nanoHUB applications.
 - Mounted volumes - allow for reduced memory requirement when loading docker container
 - nanoHUB application files sent as tarballs and are not removed at job completion to reduce bandwidth requirement
 - User supplied data is also sent as a tarball but is job specific and is removed at job completion

BOINC - APPROACH

- **BOINC Server**

- stage_file - stage application tarball files in download directory.
- docker build (~15 images)
- stage_docker_image - combines docker save and stage_file to place tarballs in the download directory. Also creates nanoHUB specific vbox_*, boinc_app_*, and submit configuration file.

BOINC - APPROACH


- **Submit Server** - common interface between local and remote resources
 - submit_api.py - interface to BOINC remote job submission (https)
 - Set of standard submit scripts for each batch system
 - receiveinput.sh
 - createBatch.py
 - uploadFile.py
 - uploadFiles.py
 - submitbatchjob.sh
 - submitBatchJob.py
 - submitBatchJobs.py
 - transmitresults.sh
 - fetchBatchOutput.py
 - cleanupjob.sh
 - retireBatch.py
 - killbatchjob.sh
 - abortBatch.py

BOINC - APPROACH

- **nanoHUB Application**
 - Scientific code remains the same
 - Job execution - adds/changes submit venue to boinc



nanoHUB/BOINC - JOIN

- **Point BOINC client to** - https://boinc.nanohub.org/nanoHUB_at_home
 - **General information** - <https://boinc.berkeley.edu>
 - **boinc2docker** - <https://github.com/marius311/boinc2docker>
 - **docker** - <https://www.docker.com>
 - **VirtualBox** - <https://www.virtualbox.org>
- 

A dark gray background with a complex network of light gray lines and dots, resembling a globe or a data network. The lines connect various points, creating a dense web of connections.

QUESTIONS & ANSWERS

?