

# Architecture

## ASRs

Identifying architecturally significant requirements:

- Specification of experiments
  - Program(s) to be run
  - Input
  - Neronet should not create restrictions to the creation of experiments
  - Should be streamlined and simple
- Management of queues
  - Work with slurm, jobman or other
- Batch submission of experiment jobs to computing clusters
  - Shouldn't overload the clusters
  - Should check clusters resource availability
- Monitoring of ongoing experiments' logs and parameter values
  - Real time access
- Access to experiment information during and after the run
  - Output values must be stored
- Configurable
- Configurable notifications on experiment state and progress
- Configurable criteria for experiment autotermination
  - Input must have a specific format
- Logging of experiment history
  - Database noSQL/SQL
- Preferences configuration

## System

### Components

#### Neroman

- The component that takes input from the user and is run on the researcher's local machine
- User specifies the experiments to be run and gives the information to Neroman
- Neroman contacts the clusters and creates and contacts Neromums

## **Neromum**

- The component created by Neroman and run on the cluster
- Creates the Nerokids, monitors them and saves the data
- Aborts failed Nerokids
- Neroman contacts Neromum from time to time to receive information on the ongoing experiments

## **Nerokid**

- A wrapper for one experiment
- Sets the specified experiment running and gives runtime information to Neromum

## **Wardens**

- TBD

## **Interfaces**

- Command line or maybe GUI for the user
- Email from Neroman to user
- SSH between Neroman and Neromums
- Sockets between Neromum and Nerokids
- Neromums collaborate with the Wardens

## **Functions by components**

- Researcher specifies experiments and data (basically specifies Nerokids)

## **Neroman**

- Start Neroman
  - Neroman reads configurations
- Connect Neroman to cluster(s) (Dispatch Neromum?)
- Display available clusters
- Display experiments
- Create experiments(s)
  - User specified experiment to be run and the data

- Multiple experiments?
  - Multiple data?
- Start experiment(s)
  - Check available clusters and cluster suitability
  - Contact clusters Neromums
- Neroman asks neromum for updates from time to time
- Neroman Emails the user at user defined moments

### **Neromum**

- Neromum is installed to cluster systems by neroman
- Neroman starts Neromum
- Neromum installs and starts the Nerokids according to the information received from Neroman (possibly with the help of a Warden)
- Neromum listens to the Nerokids and saves the information on a hdf file
- Neromum gives information to Neroman upon request
- Neroman tells Neromum to stop when the experiments are done and all data has been transferred

### **Nerokid**

- Nerokid is created by Neromum and the experiment to be done is specified
- Nerokid runs the experiment and sends information to Neromum