

## BACKGROUND – REMOTE LABS

- Laboratory work is an essential part of education in science and engineering
- Remote Laboratories are a class of systems which allow virtual access to laboratory infrastructure and learning environments through an online interface
- This eliminates the need for physical presence in the laboratory and offers a variety of logistical and economical advantages over traditional laboratories including:
  - Unlike online simulations, a remote laboratory gives real-world data from real experiments offering greater insight than would otherwise be available with *idealized* data
- The UTS Remote Laboratory system uses a partially open-source Remote Laboratory management framework known as Sahara which as yet does not support collaboration between students



UTS Remote Labs Hardware

## MOTIVATION - COLLABORATION

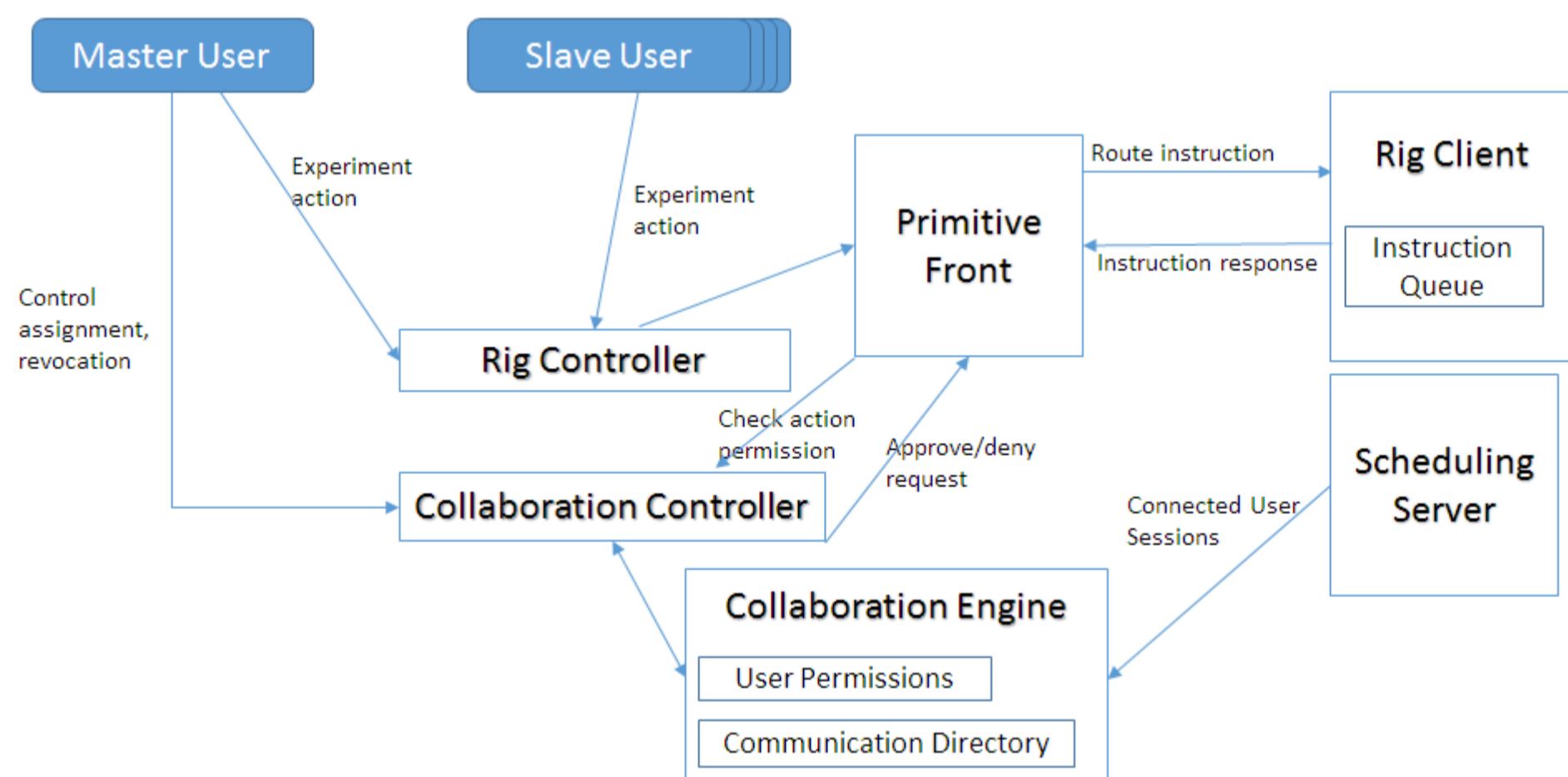
- Group work is an important part of any science or engineering curriculum
- Any laboratory environment needs to have provision for collaboration both in the context of student-student interactions and teacher-student interactions
- The types of social interactions that occur in a laboratory setting can essentially be broken down into 2 main categories:
  - Communication – Discussion of the experiment, methodology, results, etc. This communication can occur textually, visually, or verbally
  - Control management – In most cases no more than 1 student should be performing an action on an experiment at any given time. In traditional laboratories, this is managed either by the teacher or a nominated group leader

## AIM – ADAPTING SAHARA

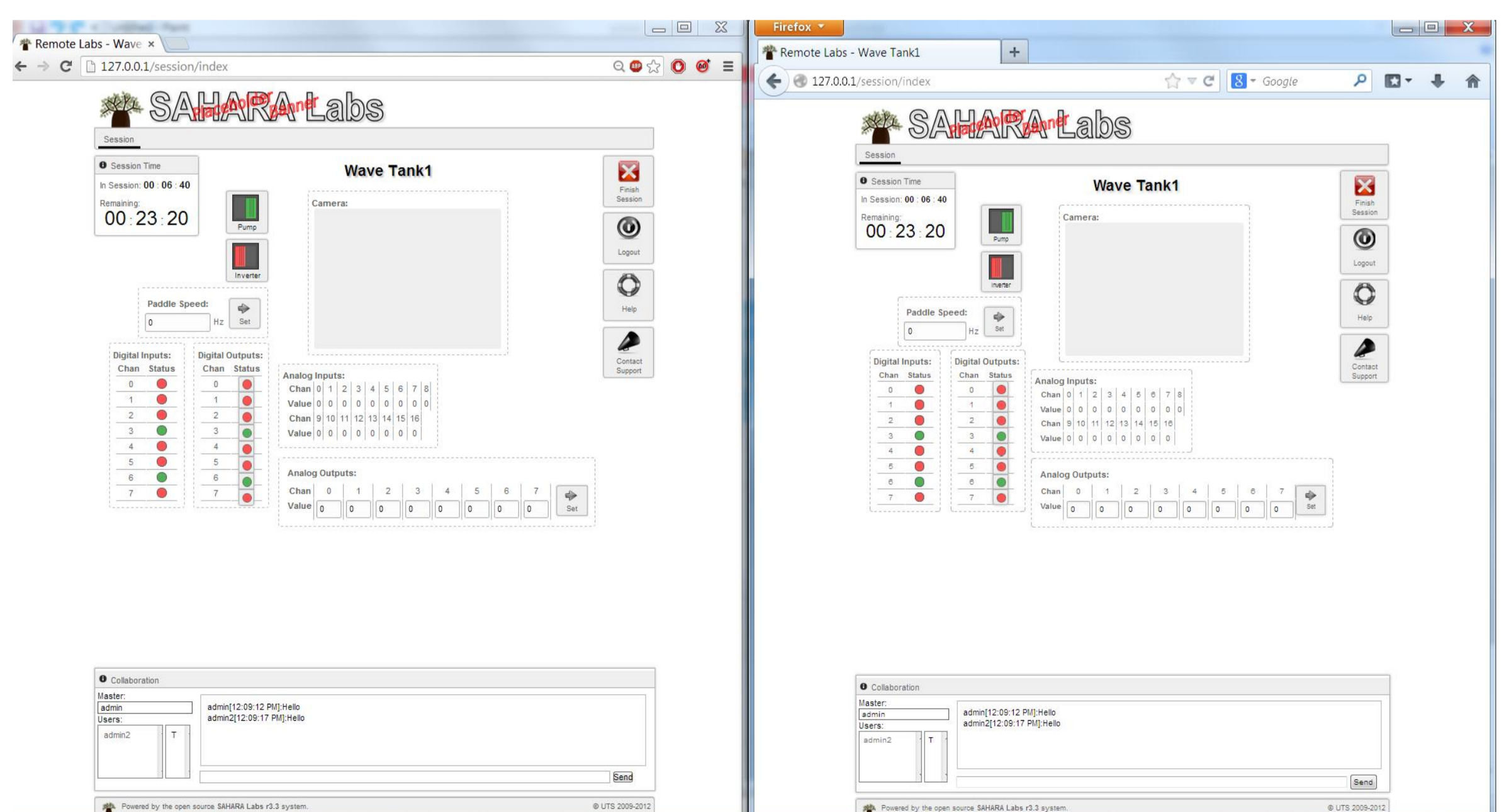
- The aim of this project is to adapt the Sahara framework to support collaboration between students
- This means providing the means for users to perform the types of social interactions that occur in physical laboratories (defined above) in an online remote laboratory

## DESIGN

- The process used for the development of the new system was RAD – additional functionality was added to the system in stages, starting with the base requirement of having more than 1 client connected to a session and then progressively adding all the functionality required to make the online environment match a real laboratory environment
- The first part of the process for adapting Sahara was to implement the functionality to allow slave user access to a rig client session
- Slave user access allowed multiple users to connect to a given session by allowing the rig client and scheduling server to identify when multiple users connect to a single session but instructions sent to the experiment hardware needed to be synchronized to ensure execution of one user's instructions didn't interfere with another's
- An instruction queue was set up on the rig client and used to synchronize the execution of instructions and prevent race conditions
- There still needed to be a way for the master user to manage slave users so that group leaders could assign rig control – this meant creating a Collaboration Engine module to manage user permissions
- The collaboration engine maintains a set of user permissions for all non-master users, which the master user can then access from the Collaboration controller
- The collaboration controller then modifies the user permissions based on the master user's instructions
- The primitive front was changed so that before it routes instructions from the web interface controller to the rig client, it first checks user's permissions on the collaboration engine to by the master user
- In this way, the master can act as a teacher/group-leader to assign control of the experiment and manage the level of access users connected to their session have
- Communication is also a part of any laboratory – students need to be able to discuss and reflect on the conducting of an experiment and the results so a text chat was implemented
- The communication directory module was built to support asynchronous message updates; when the user requests messages from the Communication Directory, the system flushes all of the un-received updates to the user – this means that if a user temporarily loses connection to the rig, they won't lose any updates from the system



Simplified Visualization of Interactions in Multi-user Sahara environment



Two Remote Laboratory Sessions In Collaboration Mode in the New System