



Lab 1

A First Distributed Blackboard

Distributed Blackboard

- Develop a program that runs on several machines
- Clients post to any server using a web browser
- Store all received data
- Propagate the newly received data
 - to all the other boards
 - in a peer-to-peer manner

How it will look like

A blackboard looks like this

The screenshot shows a web browser window titled "Blackboard GroupName - Chromium". The address bar displays "IP1:myPort". Below the address bar, a message says "reloading page in: 2 seconds. 33: error- undefined". The main content area has a heading "Submit to board" with a text input field and a "Submit to board" button. Below this is a section titled "Board contents" with a table:

Sequence Number	Entry
0	test

Next to the table are buttons for "Modify" and "Delete". At the bottom of the page, it says "Group members: member1@student.chalmers.se, member2@student.chalmers.se".

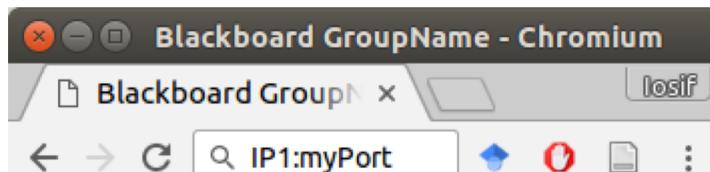
A form to submit text

Some blackboard entries

Options for each entry

A small example

Initially: empty blackboard on all vessels



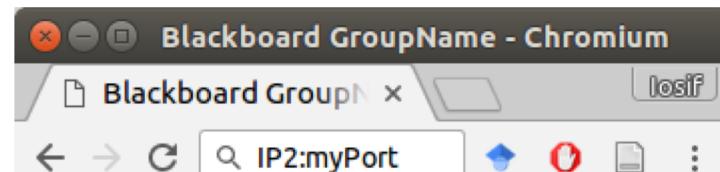
Submit to board

 Submit to board

Board contents

Sequence Number	Entry
-----------------	-------

Group members: member1@student.chalmers.se,
member2@student.chalmers.se.



Submit to board

 Submit to board

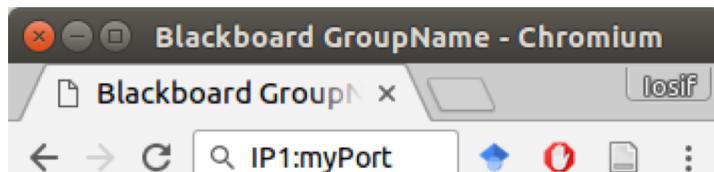
Board contents

Sequence Number	Entry
-----------------	-------

Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

A small example

On vessel 1:
Write text and submit



Submit to board

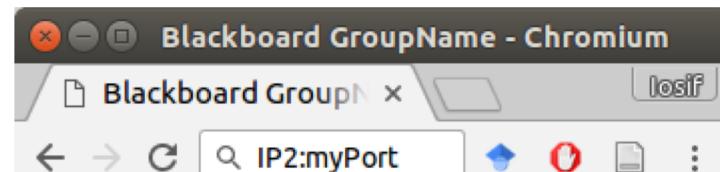


Board contents

Sequence Number

Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

On vessel 2:
No action on the browser



Submit to board

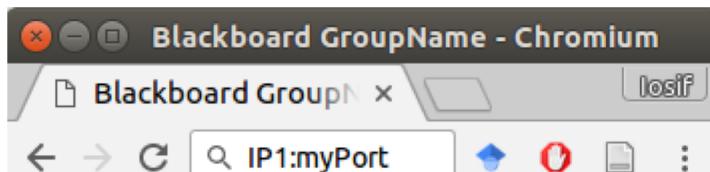
Board contents

Sequence Number

Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

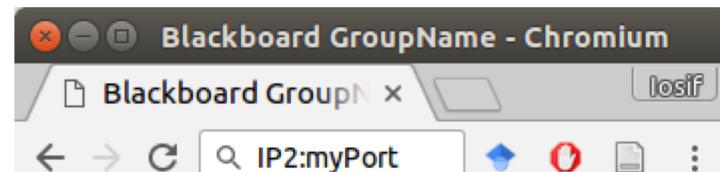
A small example

On vessel 1:
Text appears on the board



Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

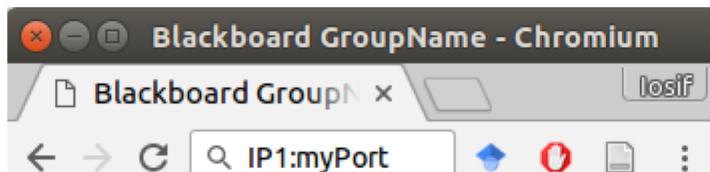
On vessel 2:
No action on the browser



Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

A small example

On vessel 1:
No action on the browser



33: error- undefined

Submit to board

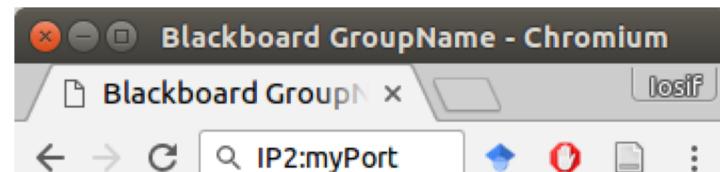


Board contents

Sequence Number	Entry
0	test Modify Delete

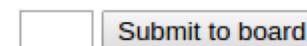
Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

On vessel 2:
Hit refresh and see post!



39: error- undefined

Submit to board



Board contents

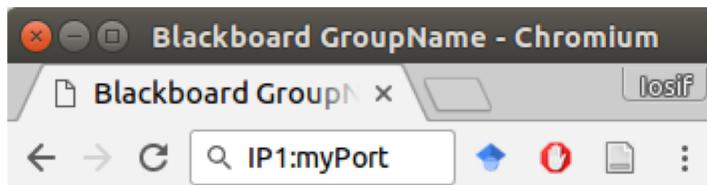
Sequence Number	Entry
0	test Modify Delete

Group members: member1@student.chalmers.se,
member2@student.chalmers.se.



A small example

A user should also be able to **modify** and a **delete** each post.



reloading page in: 2 seconds.

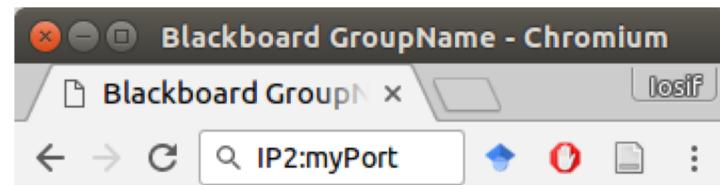
33: error- undefined

Submit to board



Board contents

Sequence Number	Entry
0	test
	Modify
	Delete



reloading page in: 3 seconds.

39: error- undefined

Submit to board

Board contents

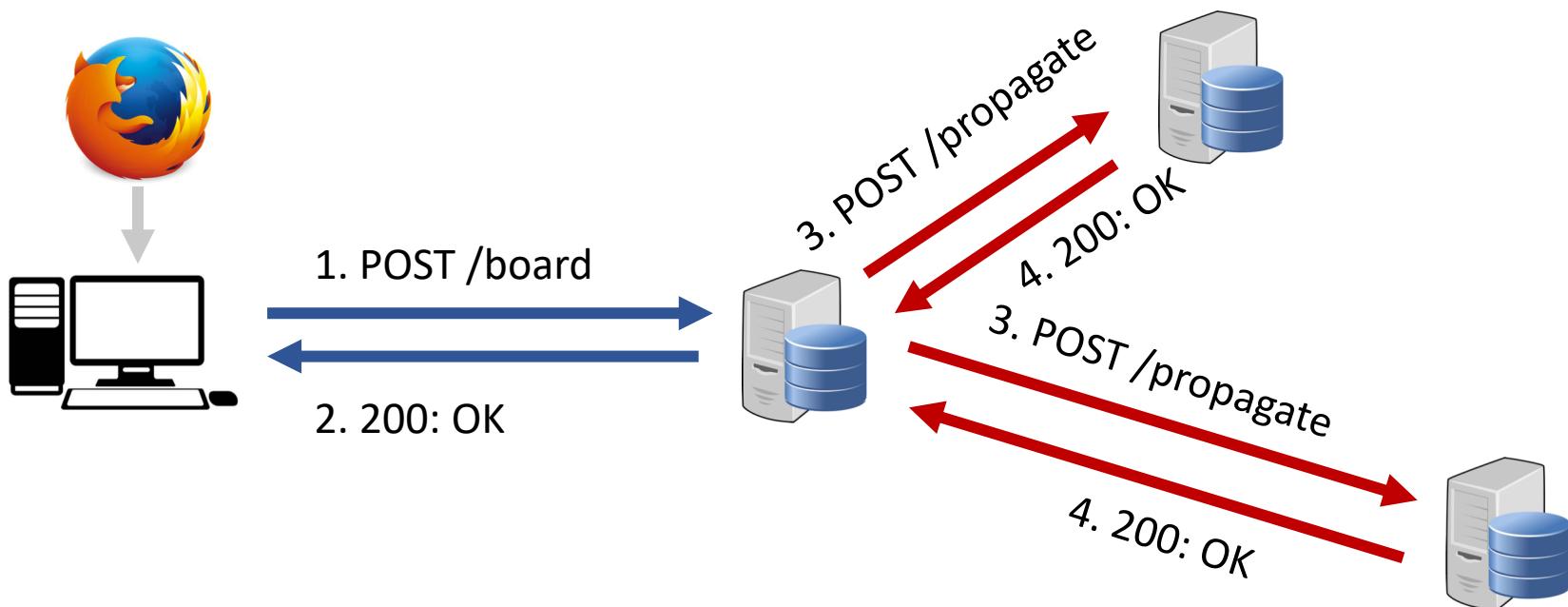
Sequence Number	Entry
0	test
	<input type="button" value="Modify"/>
	<input type="button" value="Delete"/>

Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

Communications between a client and any server

- HTTP requests
 - GET / -> display the page
 - GET /board -> display the content of the board (the data)
 - POST /board -> add a new value to the board
- HTTP Status
 - We can inform the client upon a request
 - 200: OK
 - 400: Bad Request (But this should never happen!)

Another way to view it



Code structure

- Lab configuration
 - You start the simulation with:
 - sudo python lab1.py
 - You do NOT need to change this file
- The script `lab1.py` creates several servers each call `server/server.py`
 - **You should implement your code there (`server.py`)**

What we give you

- A sample HTML file
 - Change it if you want (totally optional)
- A skeleton Python file
 - A very simple HTTP server
- A Mininet script
 - Everything is ready, run it with *sudo python lab1.py*

What you will give us

- Your code
 - Well structured
 - Well documented
- A video (or screencast) 5-10 minutes demonstrating your solution works

Lab tasks

The important stuff

Task 1: make it work

- Demonstrate that your distributed blackboard works
- Do 3 posts and show them on the other blackboards
 - Each entry has a unique number
 - All entries are propagated

Task 2: Modify and Delete values

- A user must be able to delete or modify a post
 - See the code provided
- Once a post is either modified or deleted, a node should propagate this change to other nodes