



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

Blackboard GroupName - Chromium

Blackboard GroupName x

IP1:myPort

reloading page in: 2 seconds.
33: 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](#)

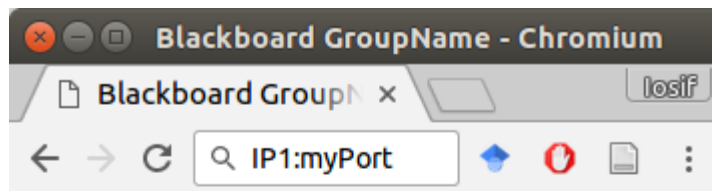
A form to
submit text

Some blackboard
entries

Options for
each entry

A small example

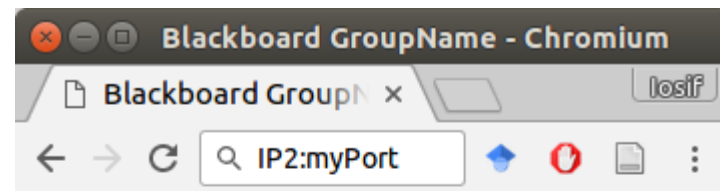
Initially: empty blackboard on all vessels



Submit to board

Board contents

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



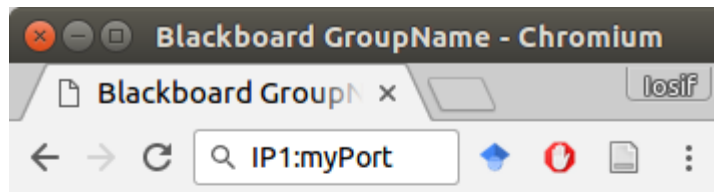
Submit to board

Board contents

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

A small example

On vessel 1:
Write text and submit



reloading page in: 0 seconds.
14: error- undefined

Submit to board

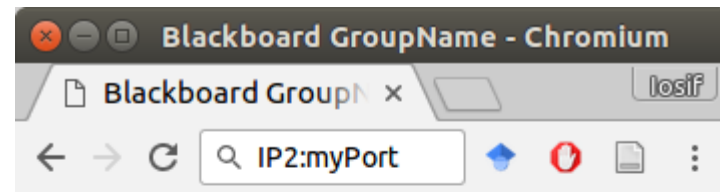
test Submit to board

Board contents

Sequence Number Entry

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

On vessel 2:
No action on the browser



reloading page in: 2 seconds.
9: error- undefined

Submit to board

Submit to board

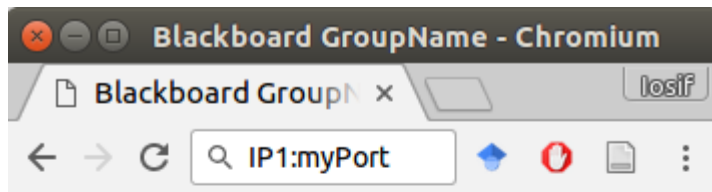
Board contents

Sequence Number Entry

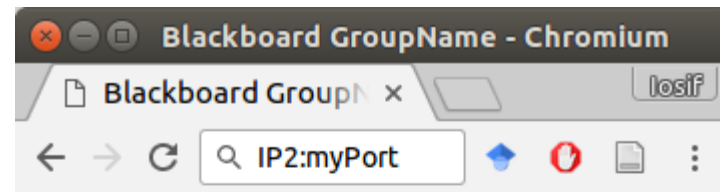
Group members: member1@student.chalmers.se,
member2@student.chalmers.se.

A small example

On vessel 1:
Text appears on the board

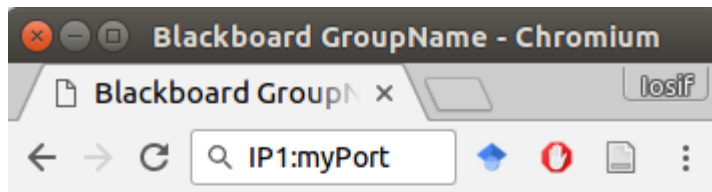


On vessel 2:
No action on the browser



A small example

On vessel 1:
No action on the browser



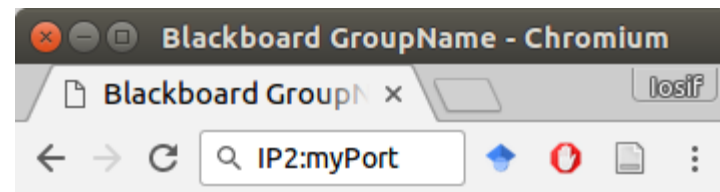
Submit to board

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.

On vessel 2:
Hit refresh and see post!



Submit to board

Board contents

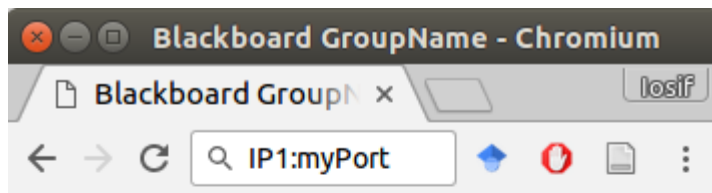
Sequence Number	Entry
0	test <input type="button" value="Modify"/> <input type="button" value="Delete"/>

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



A small example

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



reloading page in: 2 seconds.
33: error- undefined

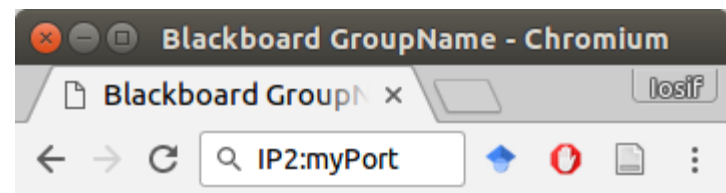
Submit to board

Board contents

Sequence Number	Entry
0	test



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



reloading page in: 3 seconds.
39: error- undefined

Submit to board

Board contents

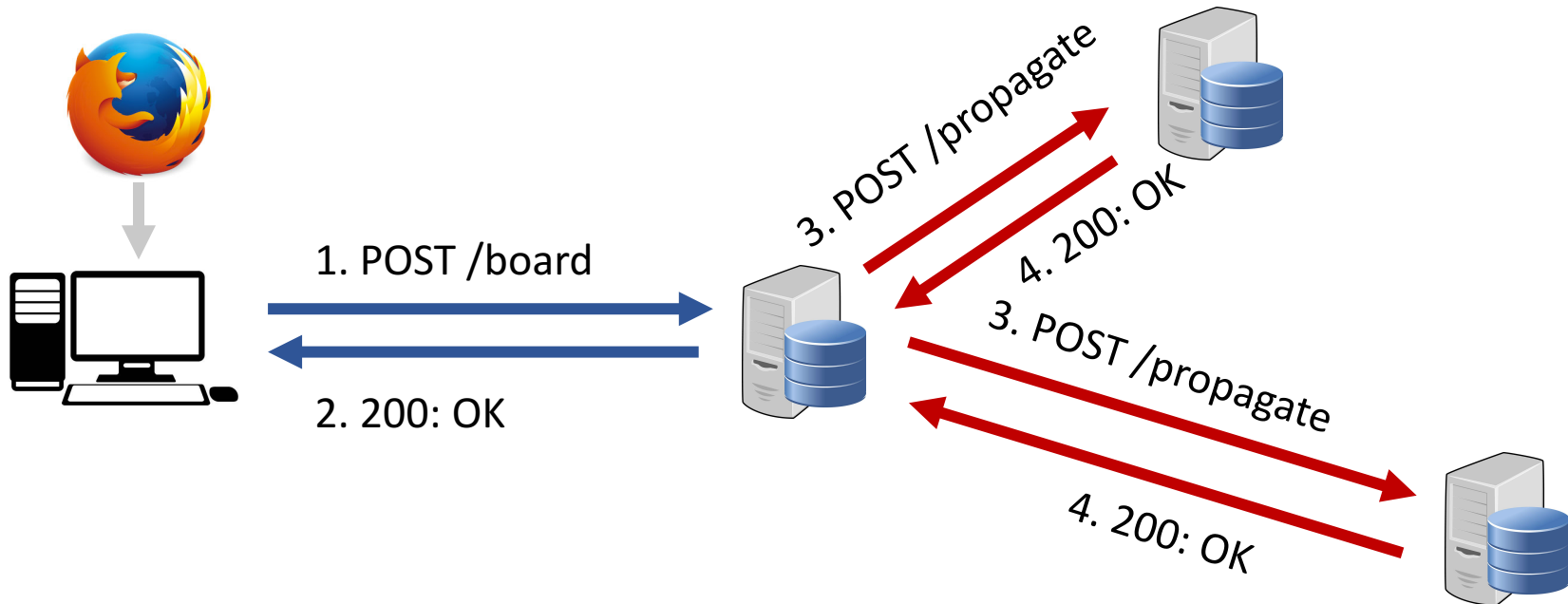
Sequence Number	Entry
0	test

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



You'll test on more servers, but the principle stays the same!

Some hints to help you

- Keep a list of all vessels in each vessel
 - (We know, this is not a scalable design, you will work on this aspect in the following labs)
- Upon a post
 - Send the update to all other vessels
 - But don't wait for the other to reply before responding to the client!
 - Store the value, it should be shown at the next refresh

Some hints to help you (2)

- HTTP formatting
 - We do not care how it looks as long it is usable
 - Don't lose too much time on formatting, 'hello world' and 'hello+world' are both valid entries
- The mininet script we are giving you directly calls a python script called *server/server.py*
 - You should implement your code there
 - The script also provides the server IP as input to the script, if you want it (and you should)

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*
- Answers to your questions in the labs
- Demo slots

What you will give us

- Your code
 - Well structured
 - Well documented
- For EACH task
 - A video or a report for each task
 - In total, 3 videos or 3 reports!
- Unit tests
 - A way to test your code easily through a script

Surprise!

- You do not need to write a report for the tasks
 - But you need to submit a video / screencast
 - either a report or a video
- Do not spend time to make it look and sound great
Good is enough.
 - Aim for 2-3 minutes per task
 - Do not waste time trying to fit in. (5 minutes is okay too).
 - The goal is to have a fun and efficient alternative to a report

Hints for videos

- We want something useful that
 - shows a demo
 - explain the algorithm
 - explain how the interesting parts of the code work
 - answer the questions we pose in the tasks
- Software hints:
 - screenomatic
 - you can borrow a microphone
 - you can use your mobile phone if your PC does not like screencast.

Lab tasks

The important stuff

Task 1: make it work

- Demonstrate that your distributed blackboard by submitting a 1-2 minutes video (or report)
 - Use at least 8 vessels/blackboards
 - Do 3 posts and show them on the other blackboards
 - Hint: show the browser windows and (optionally consoles) next to each other on your screen
 - Record your screen and document what is happening by using your mouse and your voice
 - No video editing, cutting, etc. required
 - Do not waste your time

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 vessel should propagate this change to other vessels
- Submit a 1-2 minutes video/report that demonstrates this functionality

Task 3: Is our system consistent?

- Can it happen that two vessels show different blackboards? Why?
 - Even when all data was reliably send to all vessels, and then we hit refresh afterwards
 - Hint: Use a script to do “concurrent posts” to your blackboards! (use the curl command)
- Submit 1-2 minutes video or a report
 - Explaining your thoughts
 - If you can make it happen: document it in your video or report