Kiel University
Christian-Albrechts-Universität zu Kiel

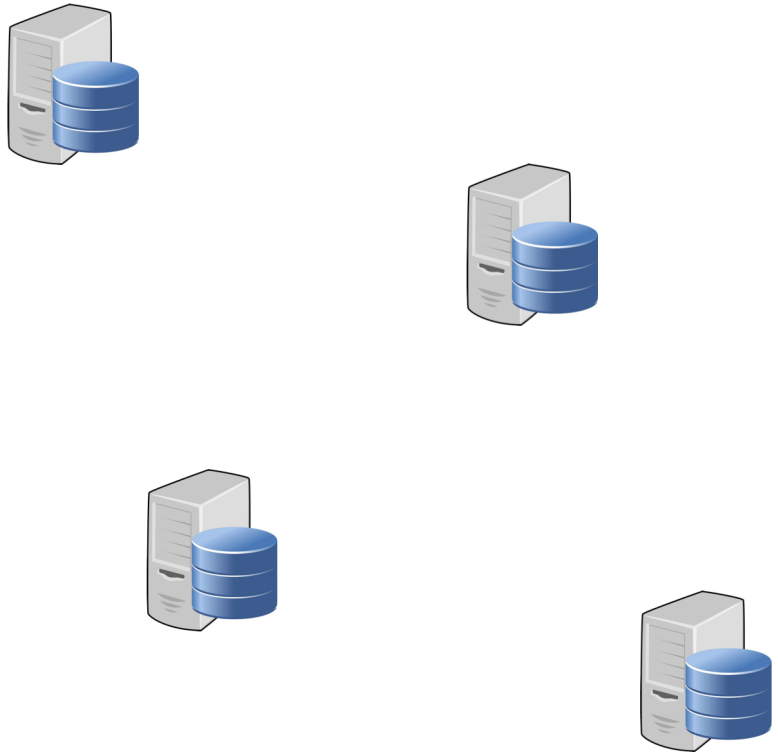
Distributed Systems
Research Group

Lab 1- Presentation

Creating your first distributed blackboard

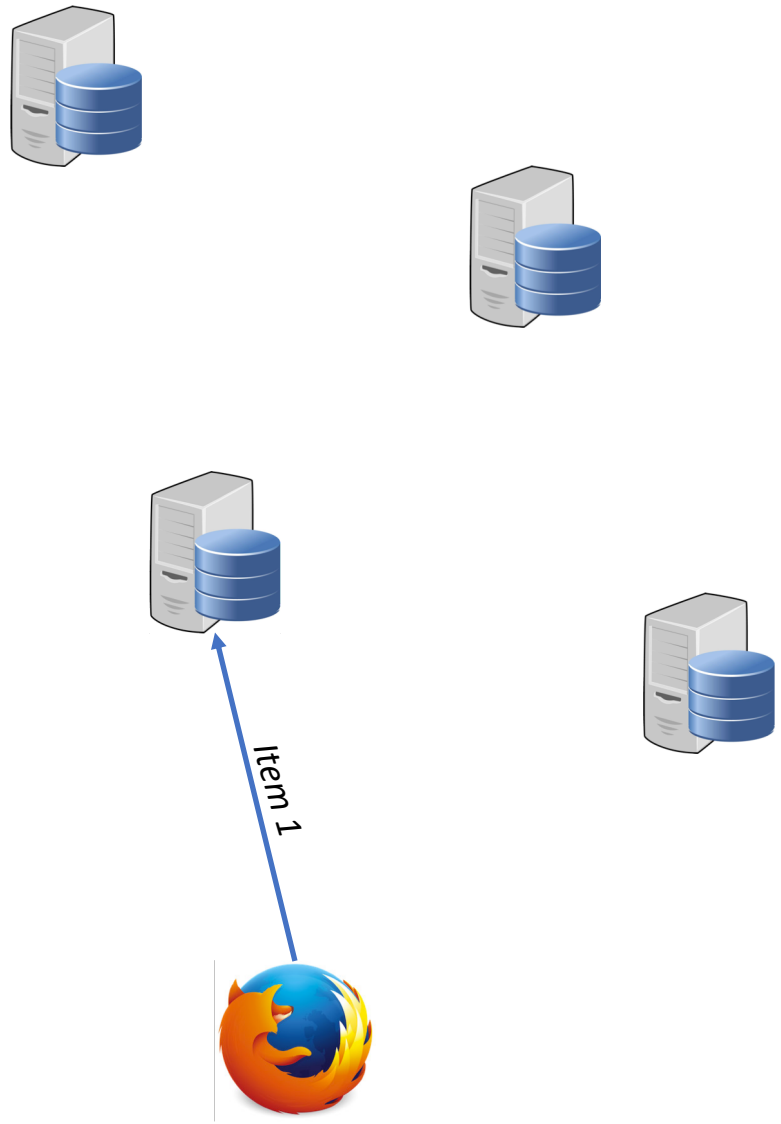
Distributed Blackboard

- A web application running on multiple (distributed) servers



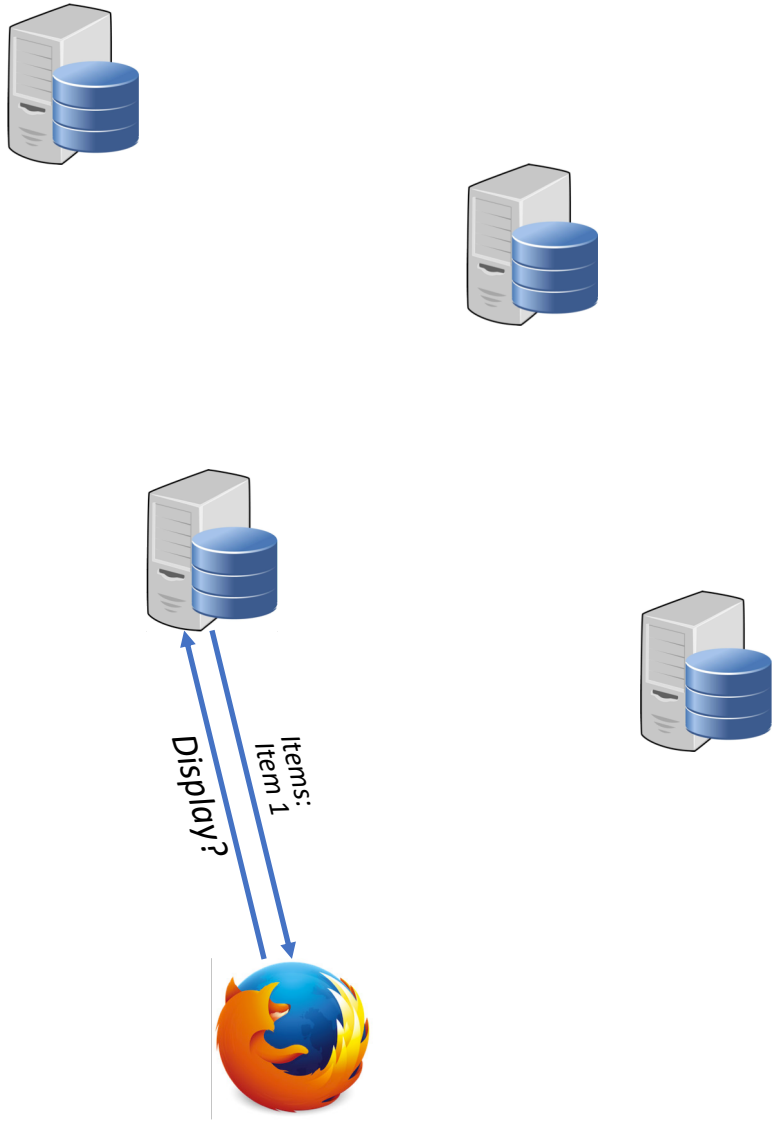
Distributed Blackboard

- A web application running on multiple (distributed) servers
- Clients can connect to any server and post information via a web browser



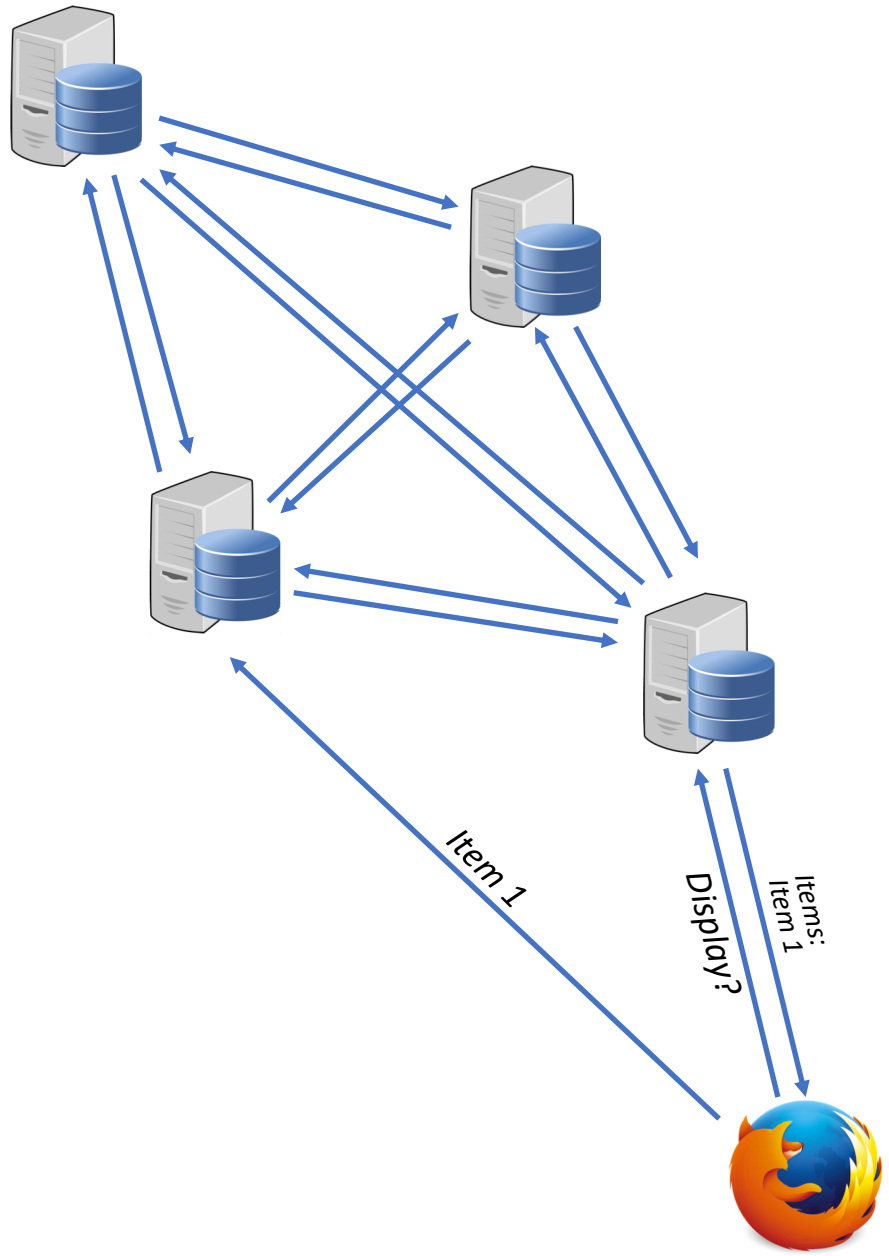
Distributed Blackboard

- A web application running on multiple (distributed) servers
- Clients can connect to any server and post information via a web browser
- Server should store all data received, and display it



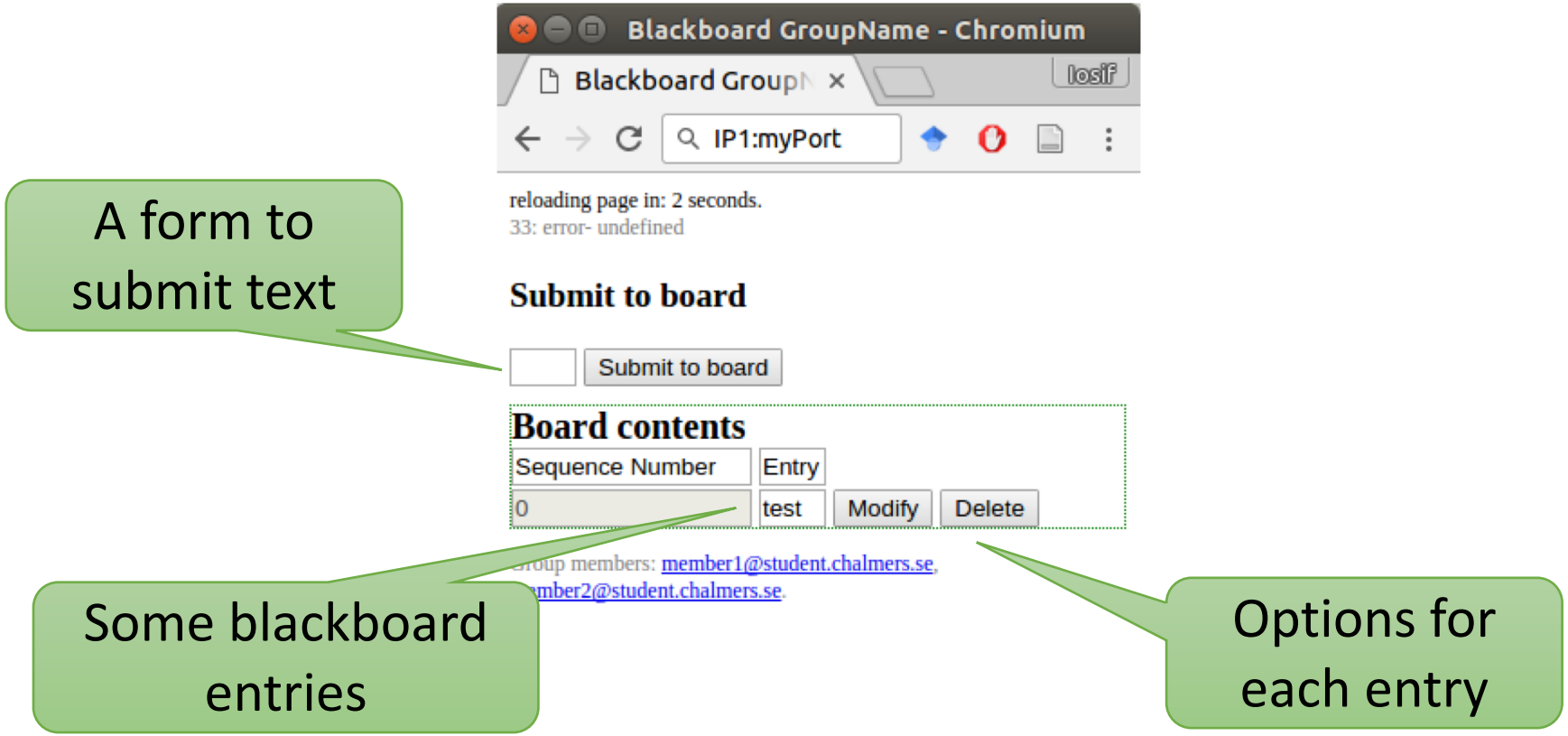
Distributed Blackboard

- A web application running on multiple (distributed) servers
- Clients can connect to any server and post information via a web browser
- Server should store all data received, and display it
- Client data must be seen from any server
  - Propagate information to all boards
  - In a peer-to-peer-manner



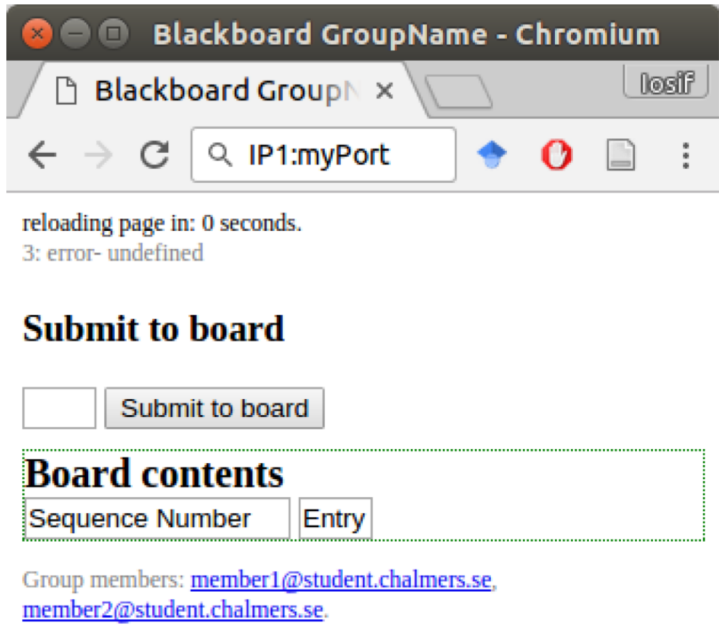
The blackboard application

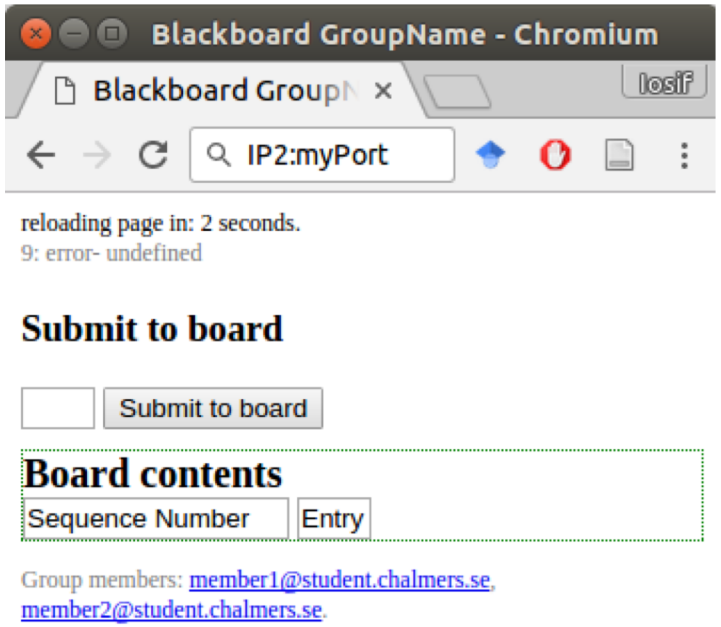
- It might look like this (or slightly better)



Example

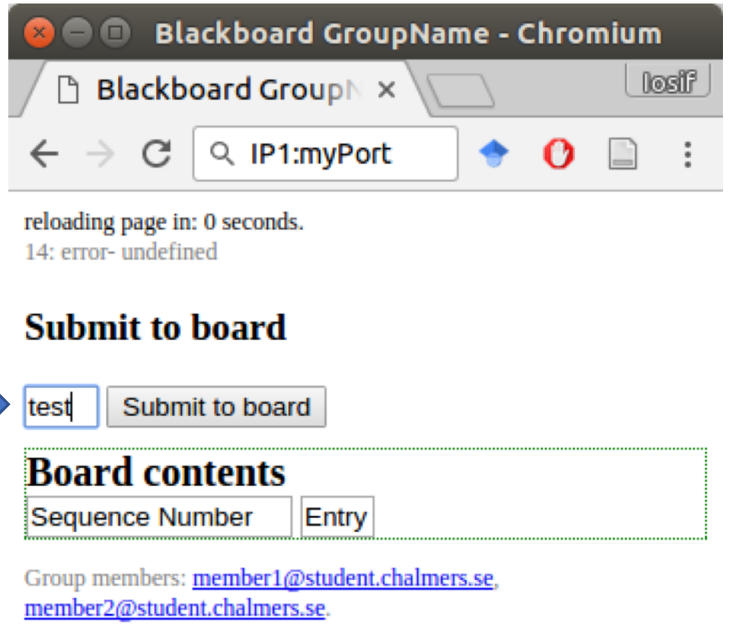
- Initially, blackboard empty on all servers



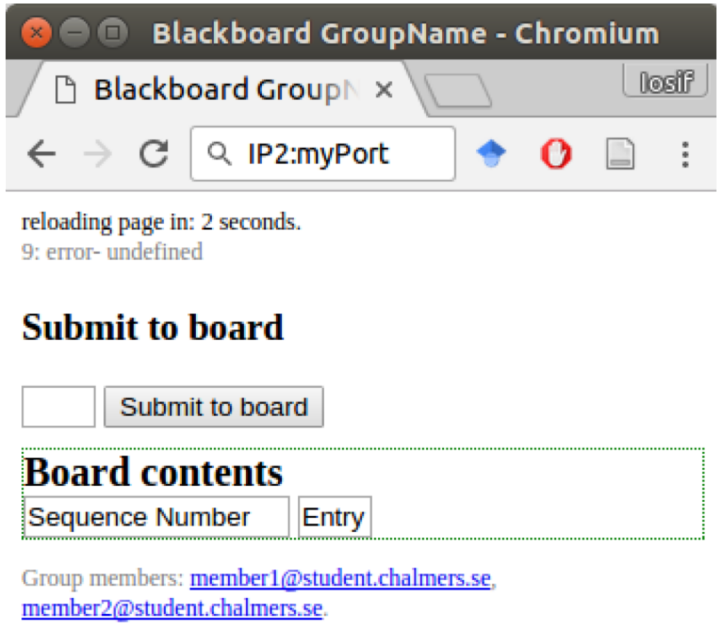


Example

On server 1:
Write text and submit

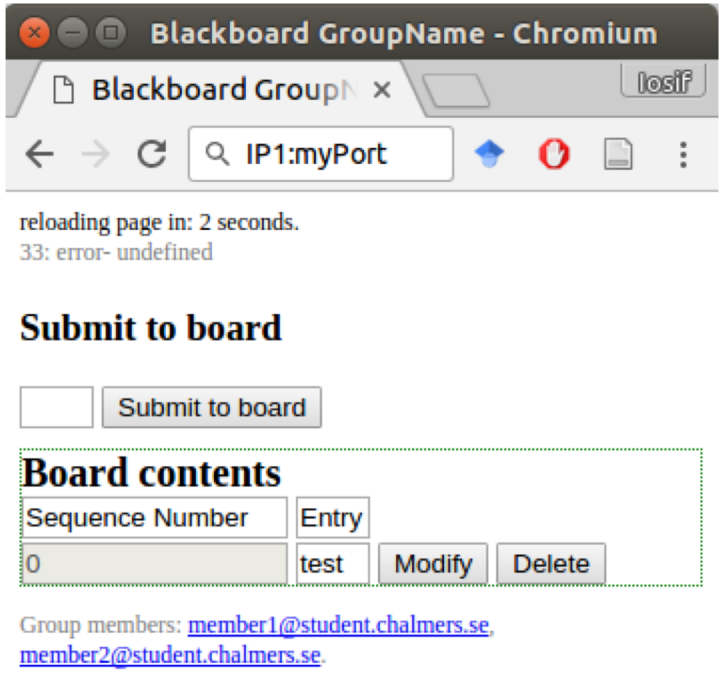


On server 2:
No action on the browser

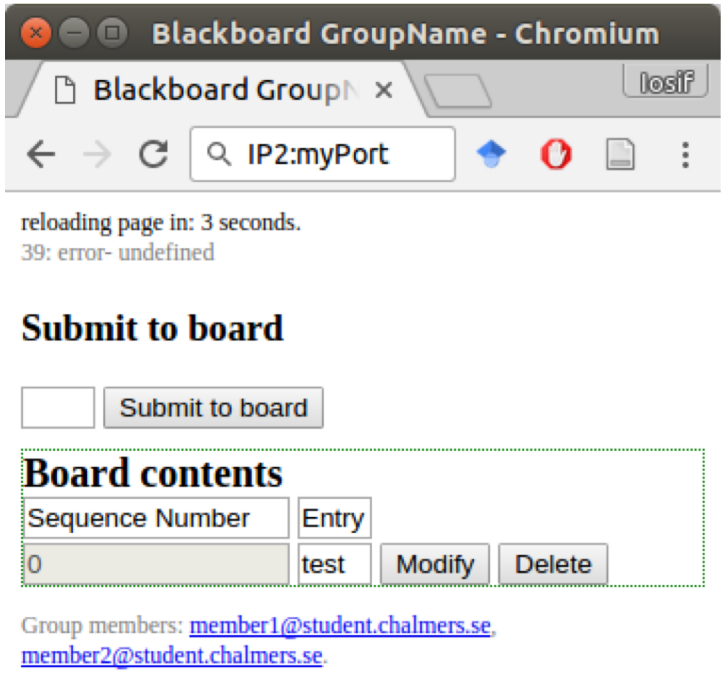


Example

On server 1:
No action on the browser

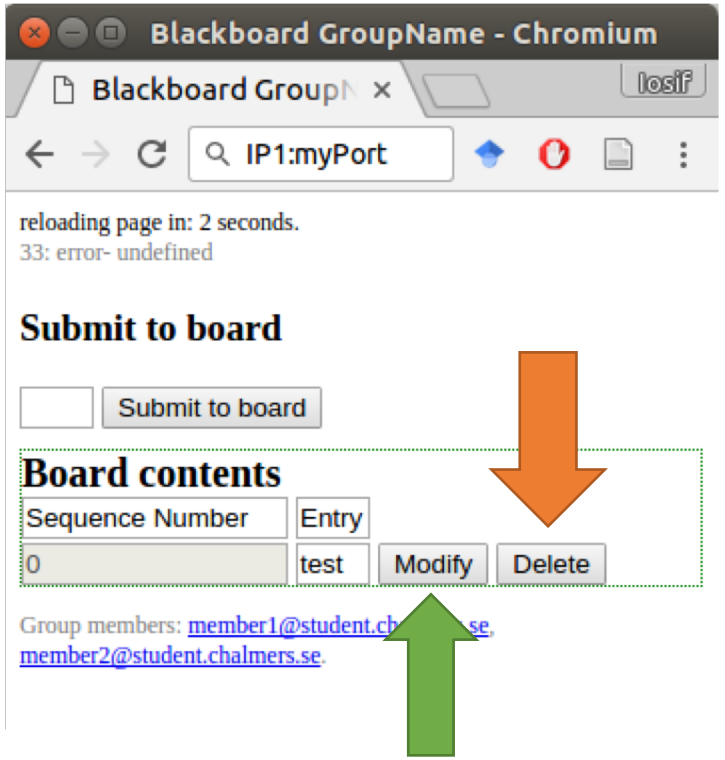


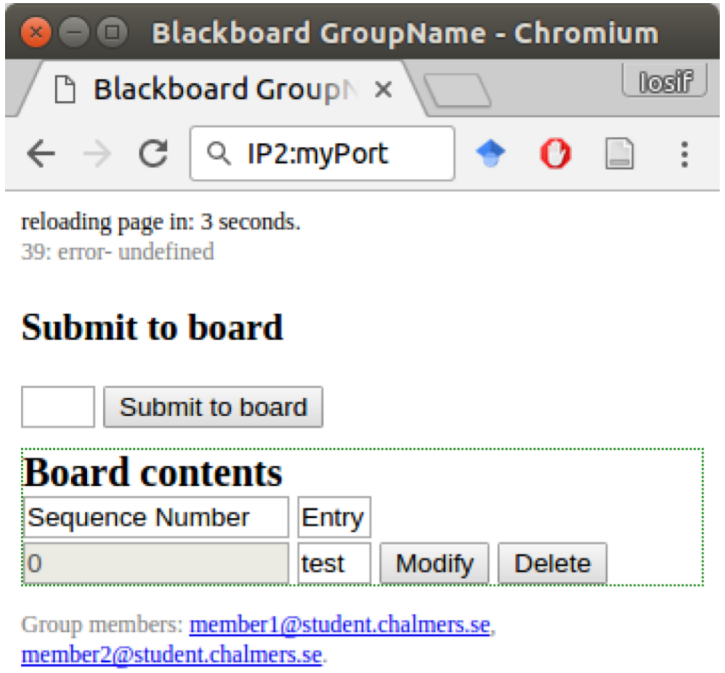
On server 2:
Hit refresh and see post!



Example

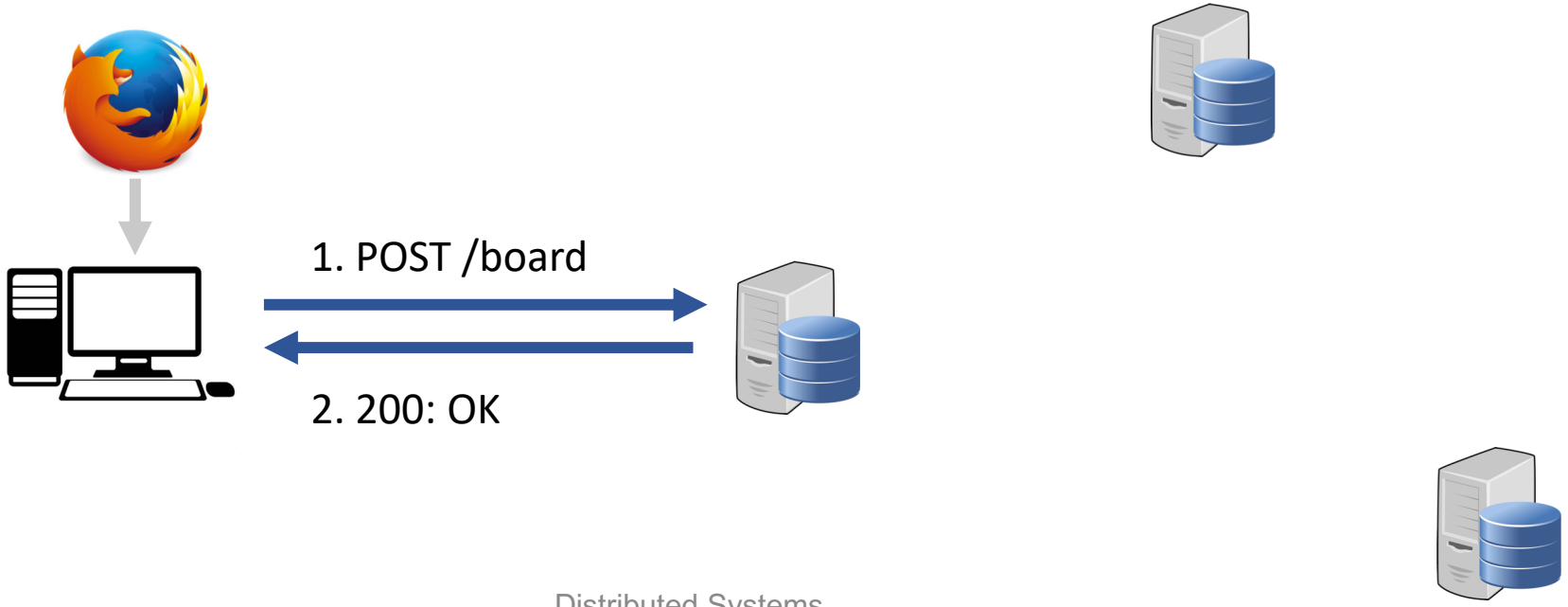
A user should also be able to **modify** and a **delete** each post.





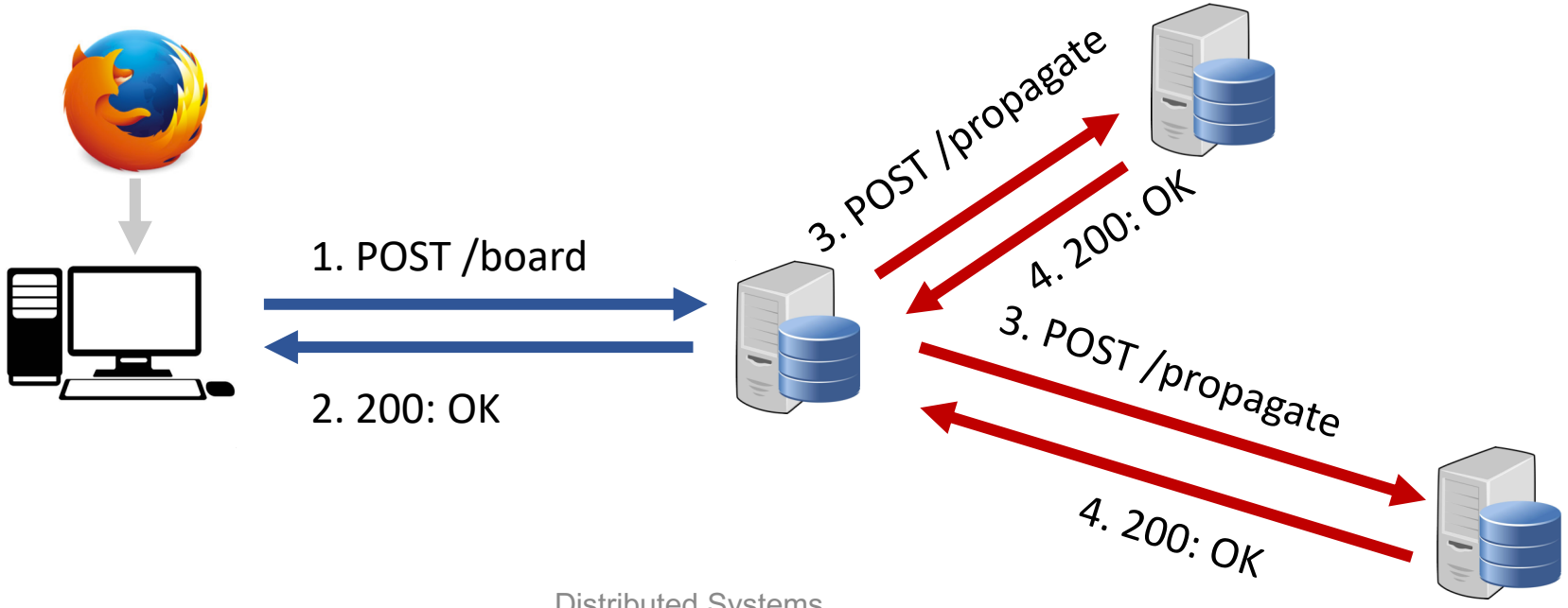
Client-Server Communication

Action	URI	Explanations	Parameters
GET	/	Display index page	
GET	/board	Display blackboard content only	
POST	/board	Add new value to blackboard	*entry*
POST	/board/<number>	Modify or delete an element from blackboard	*entry, delete*



Server-Server Communication

Action	URI	Explanations	Parameters
POST	/propagate	Propagate information	You decide!
You decide!	You decide!	You decide!	You decide!



Hints

- Keep a list of all servers locally
  - Not very scalable, but adding/removing servers is hard, trust us!

- Upon a post
  - Locally save the value
  - Propagate the data to all other servers
  - But don’t wait until you hear from all of them! This might take a lot of time, and the client is waiting for your answer!
  - Propagation should be done in parallel to responding to client

More hints

- You do not need to look at the mininet script
  - But feel free to
- Upon starting mininet, each server starts *server/server.py*
  - This is your Bottle application
- Bottle will use a template in */templates*
  - *blackboard.tpl* handles the blackboard display
  - *header.tpl* handles everything up to the board
  - *footer.tpl* handles everythong after the board

What we give you

- A Mininet script, *start_topology.py*
  - Starts the network, the servers and the server code!
  - Run it with *sudo python start_topology.py*, from the folder!

- A simple Bottle server with template
  - Your should improve *server/server.py*
  - You might have to change *server/templates/blackboard.tpl*

Kiel University
Christian-Albrechts-Universität zu Kiel

Distributed Systems
Research Group

Lab 1 -Tasks

What you NEED to implement

Task 1 – make it work

- Implement *Add* in your distributed blackboard

- Scenario:
  - Client connects to server 1, and posts 5 messages
  - Client connects to server 2 and posts 5 messages
  - Client connects to server 3 and should see 10 messages

- Requirements:
  - >= 8 servers

Task 2 – modify/delete

- Implement *Modify/Delete*

- Scenario:
  - Client connects to server 1, and posts 5 messages
  - Client connects to server 2 and modifies 1 message
  - Client connects to server 3 and deletes 1 message
  - Client connects to server 4 and should see 4 messages, with one modified

- Requirements:
  - >= 8 servers

Task 3 - consistency

- Investigate consistency
- Scenario 1:
  - Client connects to all servers and posts 5 messages to each, concurrently
  - Do server 3 and 6 show the same blackboard, in the same order?
- Scenario 2:
  - Client connects to all servers and posts 5 messages to each, concurrently
  - Client connects to all servers and modifies the same element to a different value (for each server), concurrently
  - Do server 4 and 7 show the same blackboard, in the same order?
- Requirements:
  - >= 8 servers
  - Create a script to concurrently send requests

Kiel University
Christian-Albrechts-Universität zu Kiel

Distributed Systems
Research Group

Lab 1 - deliverables

What you NEED to give/show us

Deliverables

- Your implementation
  - Including any script you used to test consistency!
  - Upload it on iLearn

- A video of your work
  - ~ 5 min (max 10)
  - Upload on the CAU-cloud
    - Everybody will be able to watch your video!
  - Should include (at least) Design – Demo – Evaluation
    - see Lab presentations for details

Deadline:
Thursday 19th, November

Good luck, and start coding!