HYPER-LINKED COMMUNICATIONS

WebRTC enabled asynchronous collaboration

Wednesday 25th May, 2016

Henrique Rocha

Instituto Superior Técnico
Universidade de Lisboa
henrique.rocha@tecnico.ulisboa.pt

Advisor: Ricardo Pereira
Co-Advisor: Paulo Chainho

OVERVIEW

1. Introduction
2. Related Work
3. Architecture
4. Implementation
5. Evaluation
6. Conclusions
7. Future Work

INTRODUCTION

CONTEXT

Written communication could never replace face to face communication.

> “No computer in our lifetimes will ever rival a human voice’s capacity to conveying rich and complex social and emotional meaning”
>
> — Geddes, Martin

Today, we can achieve more.

PROBLEM STATEMENT

Real-time communication applications can make a difference on business, education and health sectors.

An application that provides a collaborative environment and a way to remember our past communications would be a strong tool.

TÉCNICO LISBOA

THESIS GOALS

Allow multi party conference calls.

Record and playback interactive video.

Create a collaborative environment

Use only standard technologies like JavaScript, WebRTC, HTML5 and CSS3.

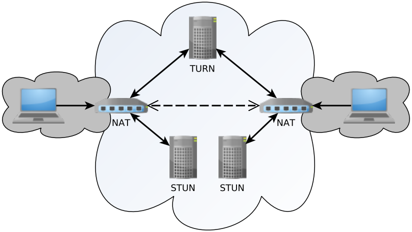
RELATED WORK

EARLY DAYS OF THE INTERNET

○ IPv4 Address Exhaustion

○ Network Address Translat

○ STUN + TURN = ICE



REAL-TIME COMMUNICATIONS

WebRTC (Web Real-Time Communications)

- Access to camera, microphone and screen*
- Peer to Peer file and stream sharing
- Standardized protocols
- No plug-ins required

* requires installing a plug-in yet.

REAL-TIME COMMUNICATIONS

Skype	Hangouts	Jitsi	Kurento
Proprietary Application	WebRTC Application*	WebRTC Application & Framework	WebRTC Framework
Audio/Video/Text	Audio/Video/Text	Audio/Video/Text	Audio/Video
File Sharing	Collaborative Tools	Collaborative Tools	Stream Recording

* requires installing a plug-in on non chrome web browsers.

HYPERMEDIA: MORE THAN WORDS, MORE THAN IMAGES

○ **Concepts:** HyperText & HyperMedia & HyperCommunications & Detail on Demand

○ **Implementations:** HyperCafe & HyperHitchcock

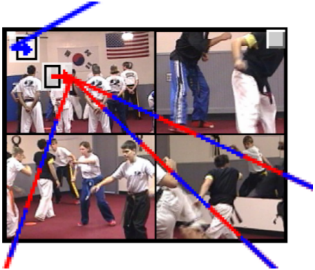


Table: Comparision between Operational Transformation libraries

Library	Own Server	Own Storage	Operations
ShareJS	✓	✓	text+objects
TogetherJS	✓	✗	text+objects
Goodow	✓	✓	text+objects
Etherpad Lite	✓	✓	extendable
OT.js	✗	✗	text

ARCHITECTURE

MODULES

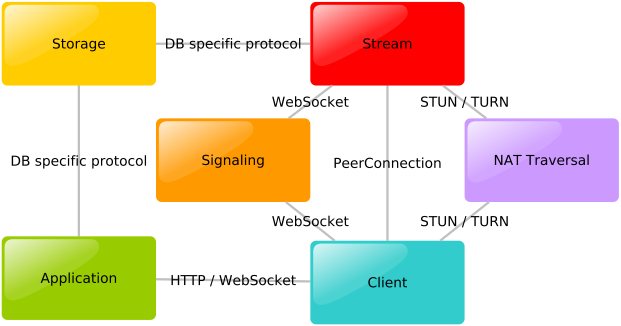


Figure: System Modules

SYSTEM INFRASTRUCTURE

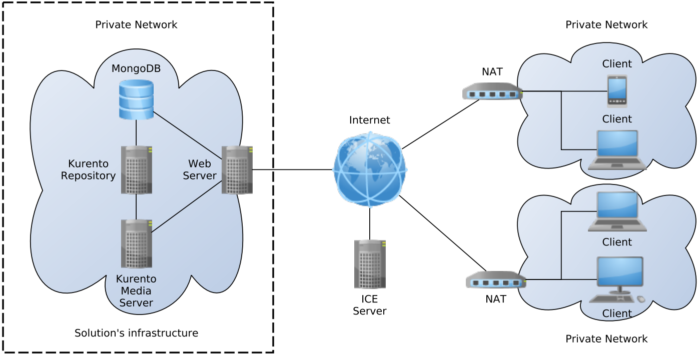


Figure: System Infrastructure

- **Signaling Server & Web Server**: Play Framework
- **Stream Server**: Kurento Media Server
- **Database**: MongoDB

APPLICATION ARCHITECTURE

Table: Application Architecture

<table>
<tr><td colspan="9">Application</td></tr>
<tr><td colspan="2">jQuery</td><td>HTML5</td><td>CSS3 (Bootstrap)</td><td>Signaling</td><td>ot.js</td><td></td><td>adapter.js</td><td></td></tr>
<tr><td>HTTP</td><td colspan="3">User Interface</td><td colspan="3">WebSocket</td><td colspan="2">WebRTC</td></tr>
</table>

IMPLEMENTATION

SIGNALING PROTOCOL

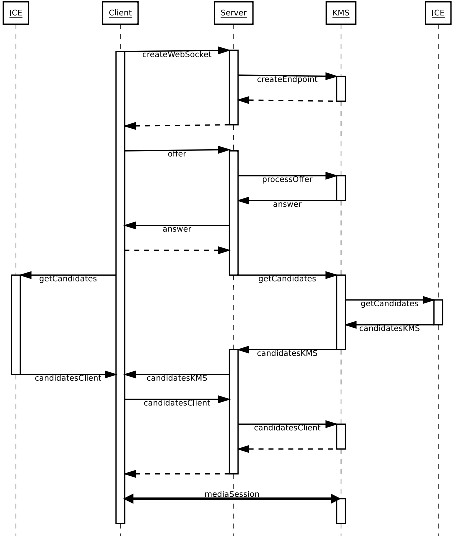


Figure: Signaling Protocol

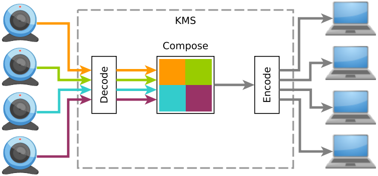
STREAM RECORDING

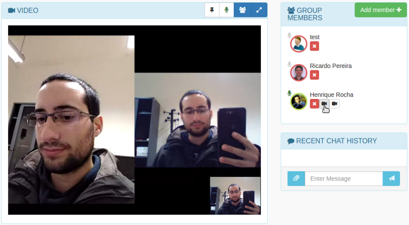
○ Client-side recording.

○ Server-side recording to file system.

○ Server-side recording to database (Kurento Repository).

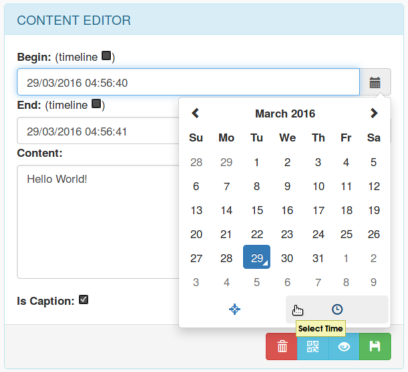
STREAM COMPOSITION





HYPER-CONTENT

- Create & Search content
- Scheduler
- QR codes
- Security concerns



CONTENT EDITOR

Begin: (timeline ■)

29/03/2016 04:56:40

End: (timeline ■)

29/03/2016 04:56:41

Content:

Hello World!

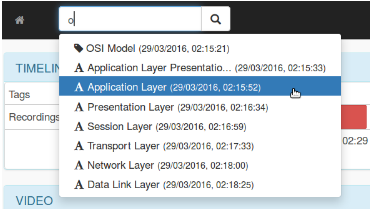
Is Caption: ☑

March 2016

Su	Mo	Tu	We	Th	Fr	Sa
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

TÉCNICO LISBOA

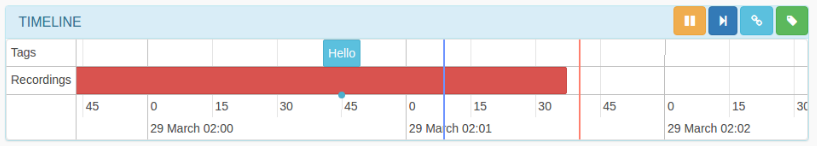
TIME MANIPULATION



o

- OSI Model (29/03/2016, 02:15:21)
- A Application Layer Presentatio... (29/03/2016, 02:15:33)
- A Application Layer (29/03/2016, 02:15:52)
- A Presentation Layer (29/03/2016, 02:16:34)
- A Session Layer (29/03/2016, 02:16:59)
- A Transport Layer (29/03/2016, 02:17:33)
- A Network Layer (29/03/2016, 02:18:00)
- A Data Link Layer (29/03/2016, 02:18:25)

- ○ Playback recordings
- ○ Create & Search annotations
- ○ Time Hyper-links



CHAT & COLLABORATIVE ENVIRONMENT

- Instant text messaging
  - WebSockets
- File sharing
  - HTTP file upload
  - stored in the database
- Collaborative text editor (OT.js)
  - retain
  - insert
  - delete

EVALUATION

PERFORMANCE TESTS - CPU

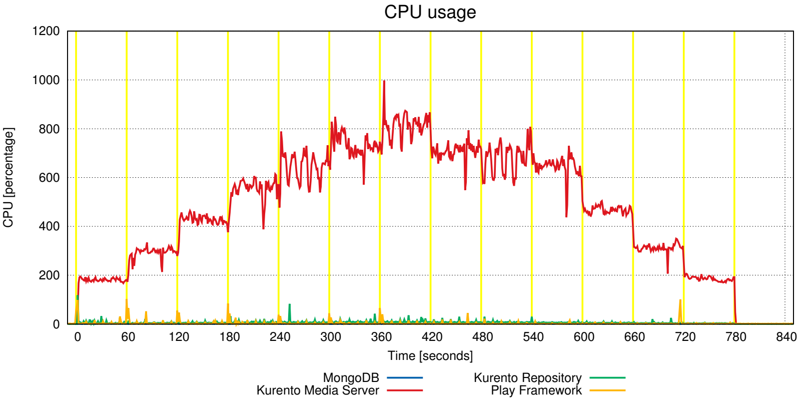


Figure: CPU usage at server

PERFORMANCE TESTS - CPU (AVERAGE)

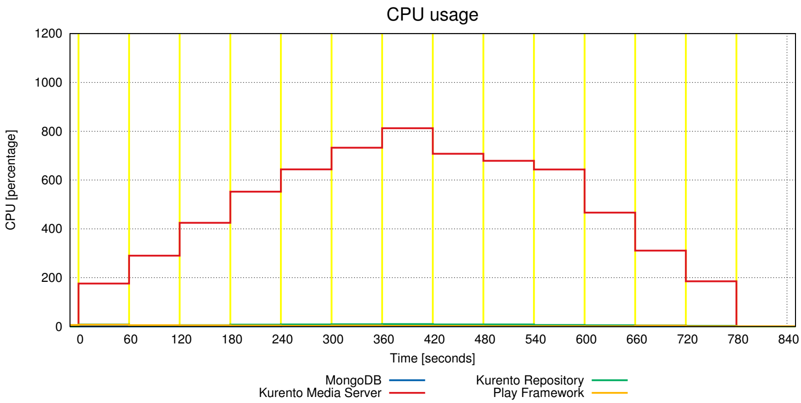


Figure: CPU usage at server (average per interval)

PERFORMANCE TESTS - MEMORY

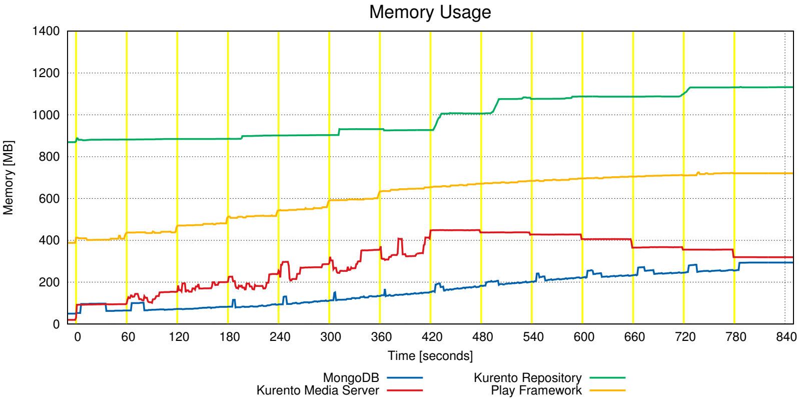


Figure: Memory usage at server

PERFORMANCE TESTS - NETWORK USAGE

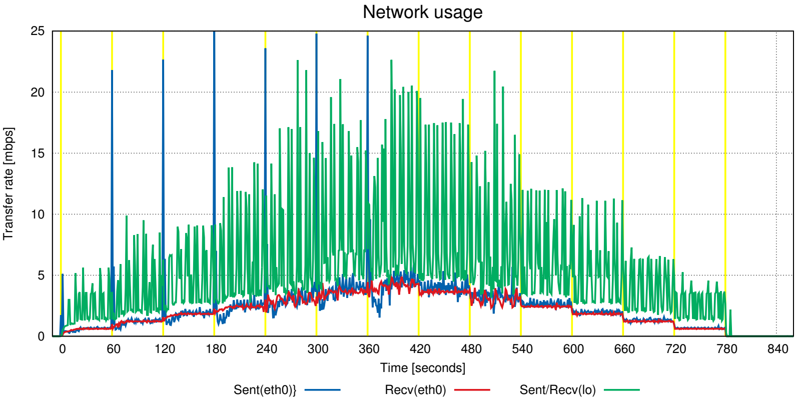


Figure: Network usage at server

PERFORMANCE TESTS - NETWORK USAGE (AVERAGE)

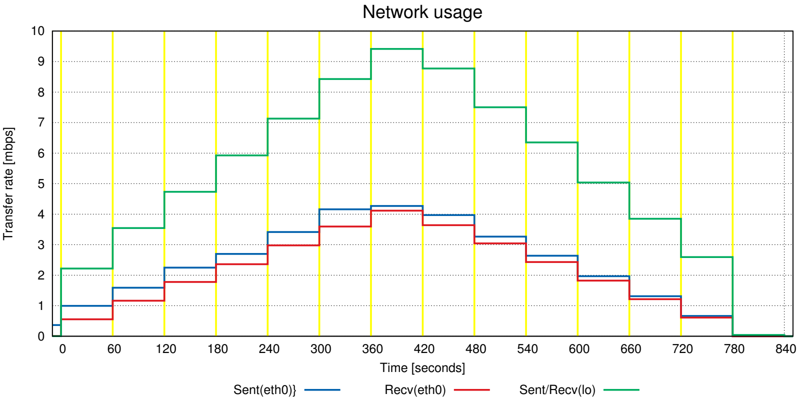


Figure: Network usage at server (average per interval)

PERFORMANCE TESTS - NETWORK USAGE

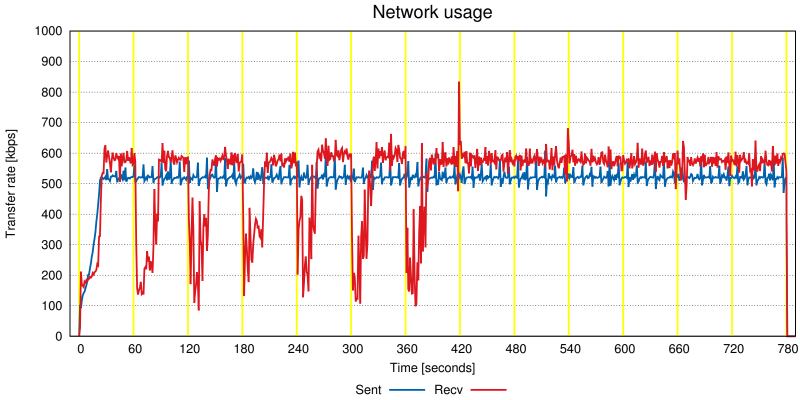


Figure: Network usage at client

USER INTERFACE TESTS

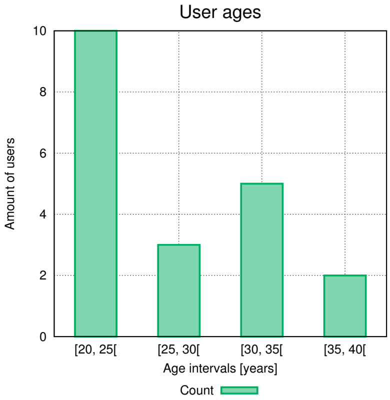


Figure: Users age

FIVE TASKS

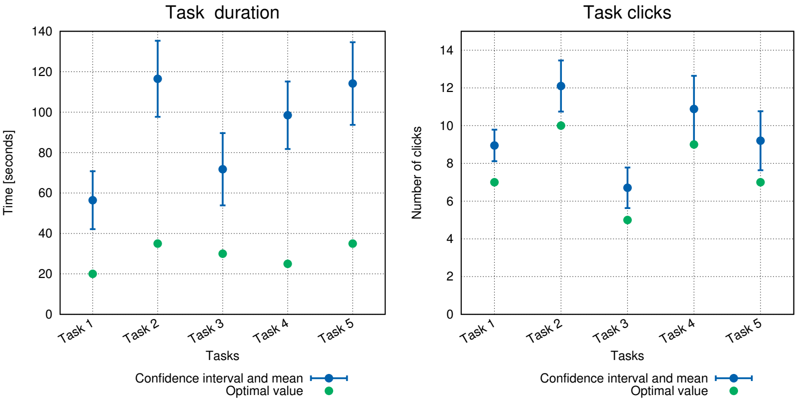
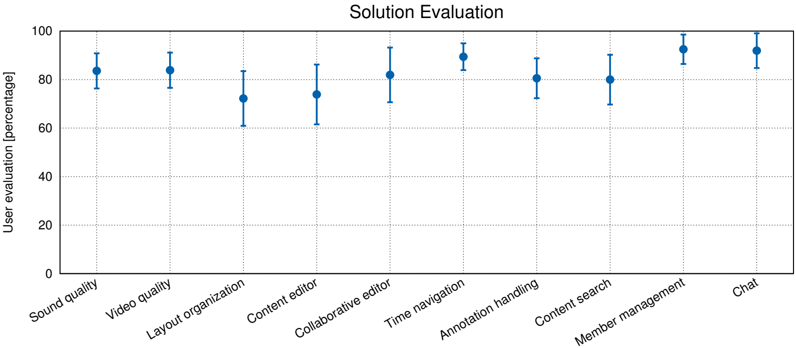


Figure: Tasks metrics

- ○ Difficulty per task.
- ○ Errors per task.

OVERALL EVALUATION



Categories

Confidence interval and mean ●

Figure: Overall evaluation

CONCLUSIONS

CONCLUSIONS

○ New usage scenarios for communication and collaboration applications.

○ Enrich communications using hypermedia concepts. Record, playback and collaboration features.

○ Prototype implementation and testing.

TÉCNICO LISBOA

DEMONSTRATION

FUTURE WORK

FUTURE WORK

- Implement fast-forward playback.
- Improve solution’s security.
- Scale our solution to multiple servers.

Questions?

TÉCNICO
LISBOA