



# Warsaw University of Technology

## Faculty of Mathematics and Information Science

# Bachelor's diploma thesis

in the field of study Computer Science and Information Systems

Design and Implementation of a Collaborative Board Using the  
Local-First Approach

**Piotr Jankiewicz**

student record book number 288767

thesis supervisor  
dr inż. Paweł Kotowski

WARSZAWA 2026



## **Abstract**

Design and Implementation of a Collaborative Board Using the Local-First Approach

The purpose of this thesis is to design and implement a collaborative board using the local-first approach. The solution uses distributed conflict free replicated data types (CRDTs) and real-time communication via WebRTC protocol to bring effortless collaboration with other users.

Final result of the effort done is an desktop application that enables drawing and erasing on the whiteboard with mutiple users simultaneously. This paper covers description of solution, but also thouches broader topic of local-first application design architectures.

**Key words:** rust language, local-first, CRDT, WebRTC, distributed system, star topology, multithread real time communication



## **Streszczenie**

Projekt i wdrożenie kolaboratywnej tablicy z wykorzystaniem podejścia „local-first”

Celem pracy inżynierskiej jest zaprojektowanie i implementacja kolaboratywnej tablicy z wykorzystaniem podejścia "local-first". Wypracowane rozwiązanie używa rozproszzonego bezkonfliktowego typu danych (CRDTs) oraz komunikacji w czasie rzeczywistym przez protokół WebRTC by umożliwić bezproblemową kolaborację z innymi użytkownikami. Ostatecznym wynikiem jest aplikacja desktop-owa umożliwiająca rysowanie oraz zmazywanie poprzednich pociągnięć na białej tablicy z wieloma użytkownikami jednocześnie. Praca pokrywa opis rozwiązania ale również porusza bardziej obszerny temat architektury typu local-first w aplikacjach kolaboratywnych.

**Słowa kluczowe:** język rust, local-first, CRDT, WebRTC, system rozproszony, topologia gwiazdy, wielowątkowa komunikacja w czasie rzeczywistym



# Contents

<b>1. Introduction . . . . .</b>	<b>11</b>
<b>Introduction . . . . .</b>	<b>11</b>
1.1. Problem Description and Analysis . . . . .	11
1.2. Other Known Solutions . . . . .	13
1.3. Motivation . . . . .	13
1.4. Examples . . . . .	13
1.5. Vision of the System . . . . .	13
1.6. Requirements Specification . . . . .	13
1.6.1. Functional Requirements . . . . .	13
1.6.2. Non-Functional Requirements . . . . .	13
1.7. Business Cases . . . . .	13
<b>2. Example chapter . . . . .</b>	<b>14</b>
2.1. Example section . . . . .	14
2.1.1. Example subsection . . . . .	14
2.2. Floats – tables and figures . . . . .	15
<b>3. The next chapter . . . . .</b>	<b>17</b>
3.1. Matrices . . . . .	17



# **1. Introduction**

## **1.1. Problem Description and Analysis**

Collaborative applications have not always been so popular as they are these days. You might not even know that some of applications that we use every day are collaborative. In everyday life of a software developer almost all of modern IDEs have collaborative work options, almost all project management software is also collaborative. There are examples close to software engineers, but these applications do not limit themselves to software development. Google Docs, Microsoft 365, Figma, Miro and many more are examples of collaborative applications that are used by millions of people every day. These applications are not only used for work, but also for education and entertainment.

It hasn't always been like this. The steep rise of collaborative applications began as the remote work became more popular but also it owes its development to universality of fast network globally and increase of computing power of everyday devices.

Current collaborative apps couldn't exist 20 years ago. They haven't been simply invented yet. Typical application of early days of online applications had a simple architecture. It was a client-server application. The server was responsible for storing data and processing it, while the client was responsible for presenting the data to the user and sending user input to the server. This architecture had its advantages and disadvantages. It was simple and easy to implement, but it had a single point of failure. If the server went down, the entire application went down. It also had scalability issues, as the server had to handle all the requests from the clients.

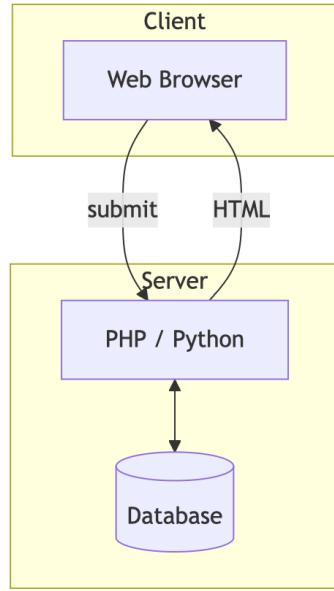


Figure 1.1: Simple client-server architecture of early internet era applications

These kind of application belong to “cloud-first” family. There was no need for “local-first” architectures yet and this term has not been popularized yet. Before even the discussion on topic of “local-first architectures” the “cloud-first” approach had its own more than one decade of fast development. From simple client server architectures the system rose to more difficult forms of large size client-side application with lots of layers and microservices. Picture below shows architectures that emerged on the market with rise of popularity of mobile and web apps from 2010.

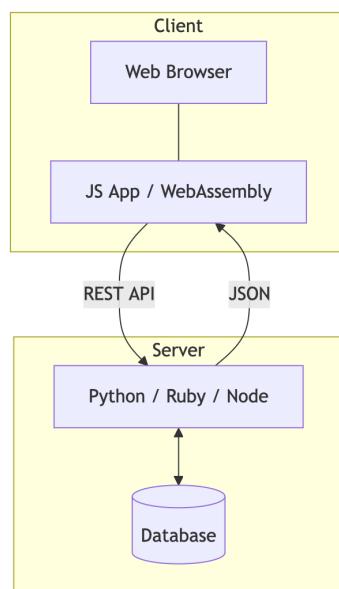


Figure 1.2: More complicated architecture of modern mobile/web applications

## 1.2. OTHER KNOWN SOLUTIONS

How far are these architectures from being good fit for collaborative applications? Let's have a look on typical modules that take part in standard user interface interaction. Picture below shows typical modules of modern web/mobile/desktop cloud-application.

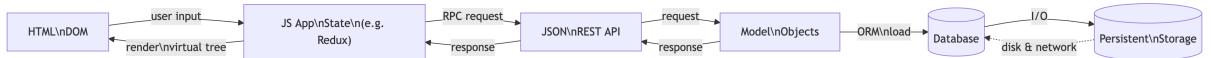


Figure 1.3: Typical modules of modern cloud-based applications

## 1.2. Other Known Solutions

### 1.3. Motivation

### 1.4. Examples

### 1.5. Vision of the System

### 1.6. Requirements Specification

#### 1.6.1. Functional Requirements

#### 1.6.2. Non-Functional Requirements

### 1.7. Business Cases

## 2. Example chapter

This T<sub>E</sub>X file is to be compiled with pdfLaTeX (it's just quick build in TeXMaker).

### 2.1. Example section

**Definition 2.1 (Definition).** A *definition* is a statement of the meaning of a term (a word, phrase, or other set of symbols).

#### 2.1.1. Example subsection

It's the deepest depth of sectioning allowed by rector.

**Definition 2.2 (Equation).** In mathematics, an *equation* is a statement of an equality containing one or more variables.

**Example 2.3.** This is an example of an equation:

$$2 + 2 = 4. \quad (2.1)$$

Equation without a number:

$$2 + 2 = 4,$$

or:

$$2 + 2 = 4.$$

It is worthwhile to peruse other mathematical environments like *multiline*, *align* and their versions with a star (, i.e. without numeration). The description of their use can be found at <https://texdoc.org/serve/amsldoc.pdf/0> starting from the end of the third page.

Equation (2.2) is false. References (and some other things) work properly after compiling T<sub>E</sub>X file twice.

$$\int_0^1 x \, dx = \frac{3}{2}. \quad (2.2)$$

Theorem 2.4 is a very interesting result.

**Theorem 2.4 (Pythagoras' Theorem).** Let  $c$  represent the length of the hypotenuse and  $a$  and  $b$  the lengths of the triangle's other two sides. Then:

$$a^2 + b^2 = c^2.$$

*Proof.* The proof has been presented in [1] and [2]. We can write then [1, 2].  $\square$

**Corollary 2.5.** The use of the term *corollary*, rather than *proposition* or *theorem*, is intrinsically subjective.

**Remark 2.6.** You can find a rather comprehensive list of available symbols at [https://www3.nd.edu/~nmark/UsefulFacts/LaTeX\\_symbols.pdf](https://www3.nd.edu/~nmark/UsefulFacts/LaTeX_symbols.pdf).

If you want to find a symbol by its shape, you can use the following site: <https://detexify.kirelabs.org/classify.html>.

**Lemma 2.7 (Someone's Lemma).** Ten lemat jest nie na temat.

*Proof.* Dowód przez indukcję.  $\square$

  Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumyeirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diamvoluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumyeirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diamvoluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## 2.2. Floats – tables and figures

Place labels after captions or you get the wrong labelling.

In Table 2.1 there are additional options for `table` and `figure` environments.

  Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumyeirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diamvoluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam non-

Table 2.1: Additional options

symbol	effect
<b>h</b>	Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot)
<b>t</b>	Position at the top of the page
<b>b</b>	Position at the bottom of the page
<b>p</b>	Put on a special page for floats only
<b>!</b>	Override internal parameters LaTeX uses for determining "good" float positions
<b>H</b>	Places the float at precisely the location in the <b>L<small>A</small>T<small>E</small>X</b> code. Requires the <b>float</b> package,[1] i.e., <code>\usepackage{float}</code> . This is somewhat equivalent to <code>!ht</code> .

Figure 2.1: Example figure – it has been drawn by **LATEX** default tools

umyeirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

### 3. The next chapter

  Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumyeirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diamvoluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumyeirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diamvoluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

#### 3.1. Matrices

Simple matrix:

$$\begin{array}{cccc} a & b & c & d \\ d & e & f & g \\ 1 & 1 & 1 & 1 \end{array}$$

Matrix with parentheses:

$$A = \begin{pmatrix} a & b & c & d \\ d & e & f & g \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Matrix with brackets:

$$\begin{bmatrix} a & b & c & d \\ d & e & f & g \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

You can also use more general environment:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix with braces:

$$\left\{ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right\}$$

**Definition 3.1.** Let  $A \neq \emptyset$ ,  $n \in \mathbb{N}$ . Every function  $f: A^n \rightarrow A$  is called an *n-ary operation* or *działaniem określonym na A*. 0-ary operations are constant functions.

**Definition 3.2 (Algebra).** The ordered pair  $(A, F)$ , where  $A \neq \emptyset$  is a set and  $F$  is a family of operations defined on  $A$ , shall be called an *algebra* (or *F-algebra*). The set  $A$  is called *the set of elements, support or universe* of an algebra  $(A, F)$  and  $F$  is called *the set of elementary operations*.

**Proposition 3.3.** I state that, having passed to the limit, the only thing left me is to camp at said limit or return, or, maybe, search for a pass or an exit to other areas.

## Bibliography

- [1] A. Author, *Title of a book*, Publisher, year, page–page.
- [2] J. Bobkowski, S. Dobkowski, Title of an article, *Magazine X, No. 7*, year, PAGE–PAGE.
- [3] C. Brink, Power structures, *Algebra Universalis* 30(2), 1993, 177–216.
- [4] F. Burris, H. P. Sankappanavar, *A Course of Universal Algebra*, Springer-Verlag, New York, 1981.

## List of symbols and abbreviations

nzw. nadzwyczajny

\* star operator

~ tilde

If you don't need it, delete it.

## List of Figures

1.1	Simple client-server architecture of early internet era applications . . . . .	12
1.2	More complicated architecture of modern mobile/web applications . . . . .	12
1.3	Typical modules of modern cloud-based applications . . . . .	13
2.1	Example figure – it has been drawn by L <sup>A</sup> T <sub>E</sub> X default tools . . . . .	16

If you don't need it, delete it.

## **Spis tabel**

2.1	Short caption . . . . .	16
-----	-------------------------	----

If you don't need it, delete it.

## **List of appendices**

1. Appendix 1
2. Appendix 2
3. In case of no appendices, delete this part.