

Narwal Auth API

MUHAMMAD NASEEM
SAMI NAEEM
KYNAT MANSHA

INTRODUCTION

The Narwhal Authentication API offers a streamlined and secure approach to handling user authentication. Utilizing Node.js for server-side operations, and JavaScript for client-side dynamics



OBJECTIVES

- Develop a secure API using ZKP for password authentication.
 - Create a demonstrative website integrated with the API.
 - Develop a JavaScript library for client-side ZKP calculations.
-

Components

C1

NodeJs Server-side

- Robust server-side operations
- Secure handling of authentication processes

C2

JavaScript-enhanced Client-side

- Responsive and interactive user experience
- Enhanced security with client-side processing

C3

MongoDB Integration

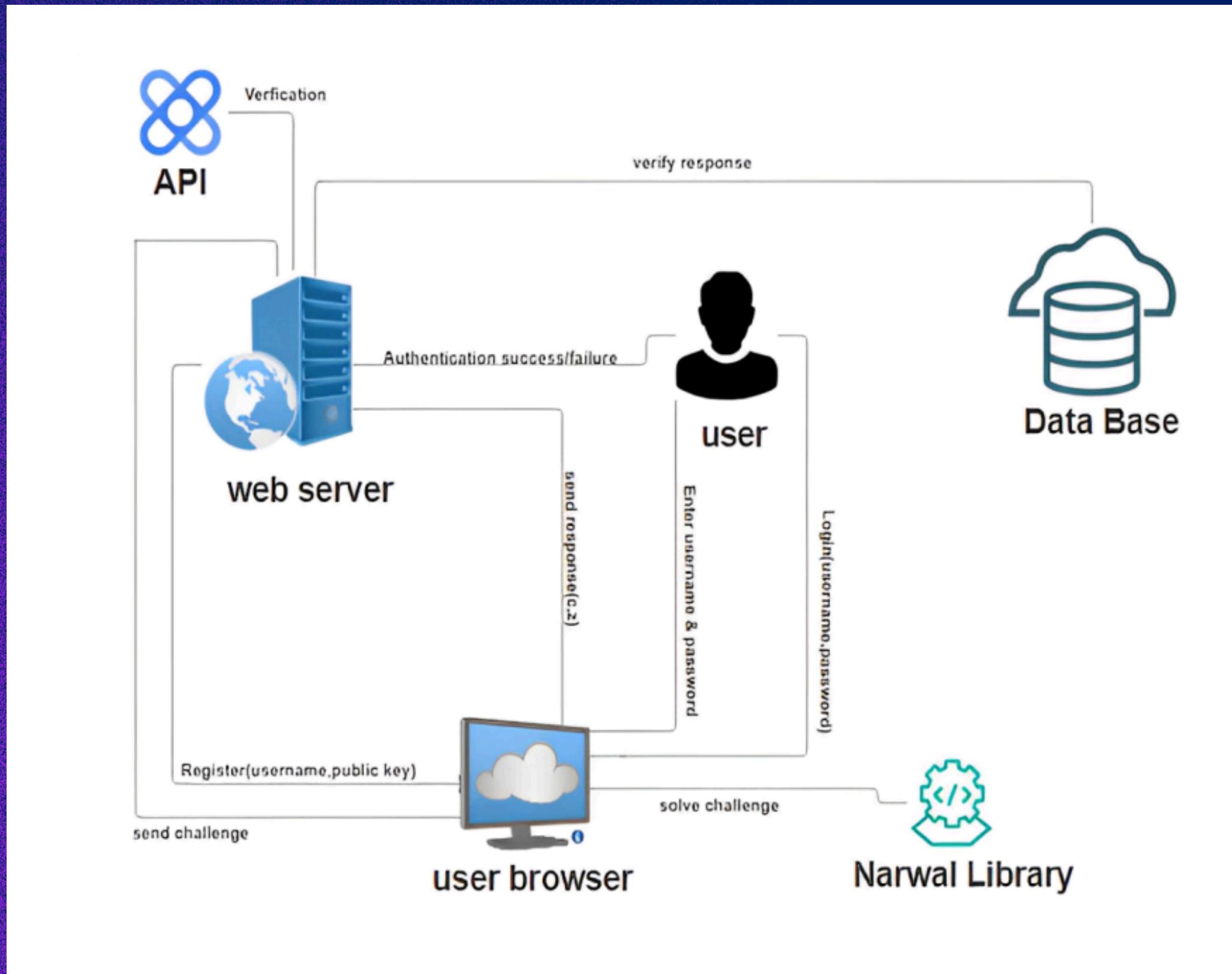
- Efficient data management
- Scalable and flexible database solution

C4

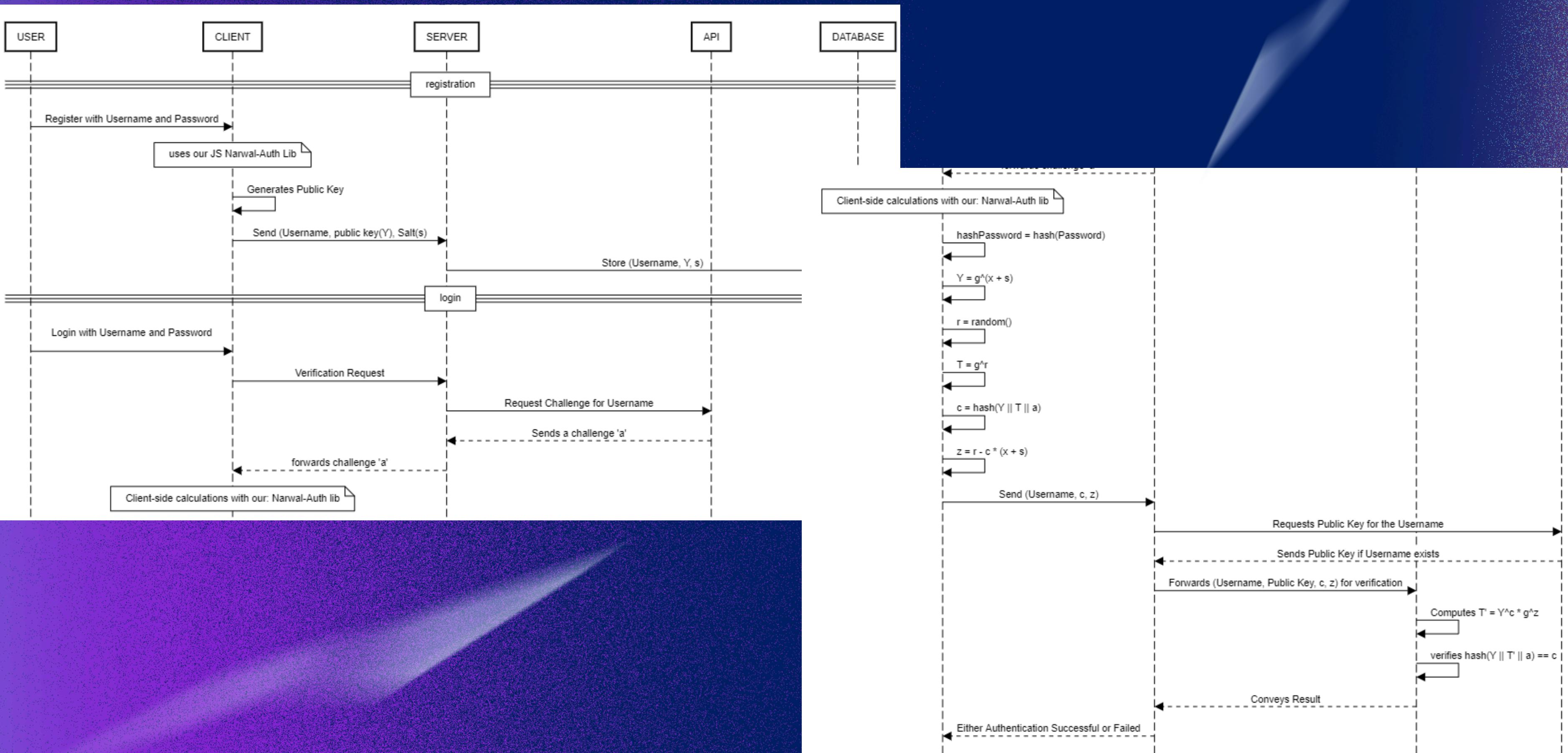
Cryptographic Methods

- Ensures secure transmission and storage of data.
- Utilizes techniques like hashing, encryption, and salting.
- Protects against unauthorized access and data tampering.

ARCHITECTURE



Sequence Diagram



BENEFITS



- 1** Enhanced Privacy
- 2** Password leaks resistance
- 3** Replay Attack immuity
- 4** Regulatory Compliance
- 5** User trust and Confidence
- 6** Scalability and adaptability
- 7** Limited data exposure
- 8** cutting-edge technology

Why It Matters

Protects Sensitive Information

- Critical in an era where data breaches are common
- Ensures personal and organizational data remains secure

Enhances User Experience

- Provides a seamless and efficient authentication process
- Reduces friction and improves overall satisfaction

Supports Business Growth

- Reliable and secure authentication fosters business trust
- Scalability ensures long-term viability and performance

Thank you