Project Name: Bride

Student : Badaruddin Khuhro, Fjorian Musaraj

Intstructor: Djordje Petrovic ,Andrew Starling

Leveraging Blockchain for Enhanced Security and Efficiency: The Bride Avalanche Capstone Initiative

1 Trillion BRIDE

Today everyone get 1000 BRIDE

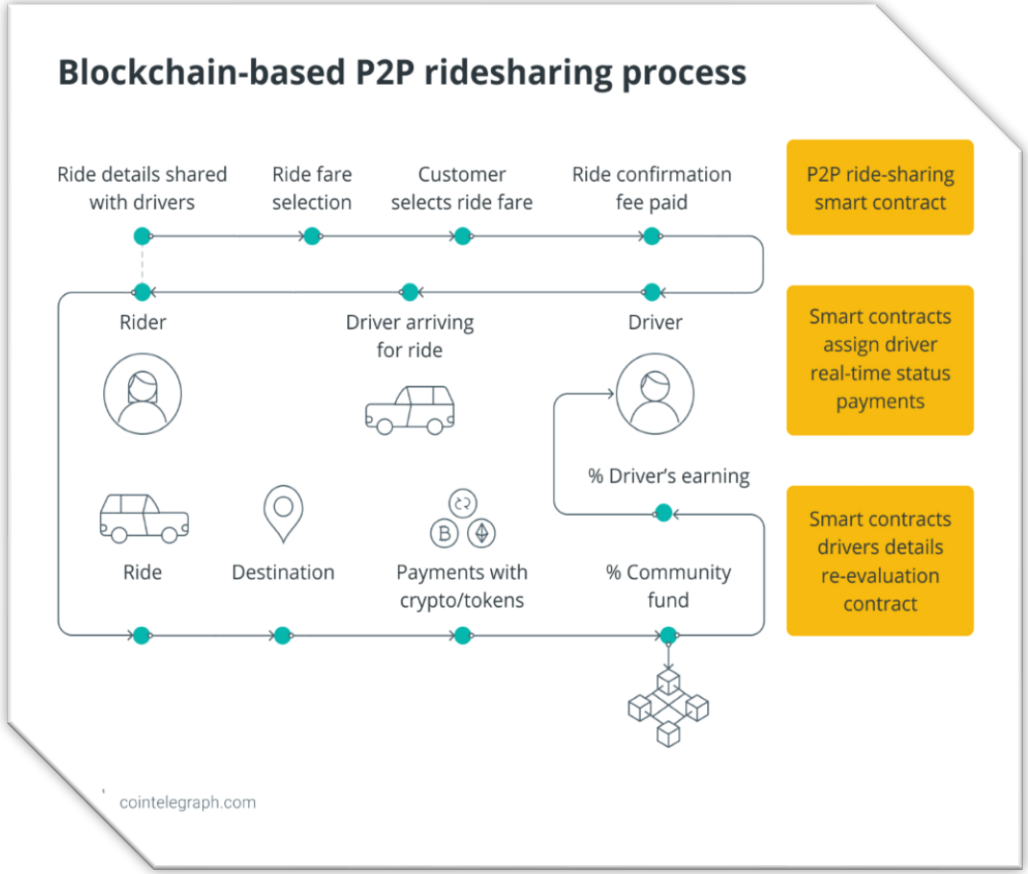
Abstract:

Contents

- This report unveils the findings of the Bride capstone project, focusing on harnessing blockchain technology, specifically the Bride Avalanche consensus protocol, to elevate the security and efficiency of the Bride ride-hailing platform. The objective was to evaluate the feasibility and potential advantages of integrating blockchain into Bride's infrastructure, focusing on data integrity, privacy, and transaction speed. The project encompassed research, analysis, simulations, and comprehensive strategies for implementation.

Introduction | 01

The emergence of blockchain technology has opened doors to transformative solutions across various industries, including transportation and logistics. The Bride capstone initiative aims to explore the potential of incorporating the Bride Avalanche blockchain protocol into the Bride platform, thereby enhancing security, efficiency, and reliability. The selection of the Avalanche protocol is based on its promising consensus mechanism, offering swift transaction confirmation, energy efficiency, and scalability.



Literature Review | 02

The literature review delves into blockchain technology, the Avalanche consensus protocol, and its potential applications in the transportation sector. Existing studies emphasize blockchain's merits, including data immutability, transparency, and decentralized control, all of which resonate with Bride's commitment to trust and reliability.

Methodology | 03

The Bride capstone project adopts a meticulous technical approach that involves multiple stages:

- **Architecture Overview:** The integration plan envisions the seamless coexistence of blockchain components within Bride's existing architecture. This amalgamation, illustrated in Figure 1, fosters efficient data synchronization and communication.
- **Smart Contracts:** Smart contracts emerge as a pivotal tool for automating and optimizing various aspects of the platform. These self-executing scripts facilitate seamless driver payments, ensuring accuracy and promptness.
- **Data Migration:** Transitioning to blockchain mandates migrating historical data while maintaining data consistency. A data mapping process ensures that existing ride and payment records seamlessly align with blockchain transactions.
- **Node Configuration:** Establishing a distributed network of blockchain nodes demands meticulous configuration. A decentralized node structure enhances system resilience, scalability, and security. Load balancing techniques are employed to ensure transaction processing is optimally distributed.

Findings | 04

Key findings from the project encompass

Transaction Speed

The Avalanche protocol demonstrates remarkable transaction speed and confirmation times, positioning it as an ideal candidate for real-time applications like ride-hailing.

Data Integrity

Blockchain's immutability significantly bolsters data integrity, reducing vulnerabilities to unauthorized changes to ride records, driver details, and payments.

Privacy

The decentralized architecture of blockchain minimizes centralized data storage, thus enhancing user privacy.

Recommendations | 05

Derived from the findings, the project suggests

Pilot Implementation

A limited-scale pilot implementation of the Avalanche blockchain within Bride's platform should be undertaken to gauge real-world performance and user experiences.

Integration Strategy

A phased approach to integration will ensure a seamless transition. Prioritizing payment processing will minimize disruption, followed by incremental inclusion of other functionalities.

Education and Adoption

Educating Bride drivers and riders about blockchain technology's benefits will address concerns and highlight its enhanced security and transparency.

Future Work | 06

The path forward involves:

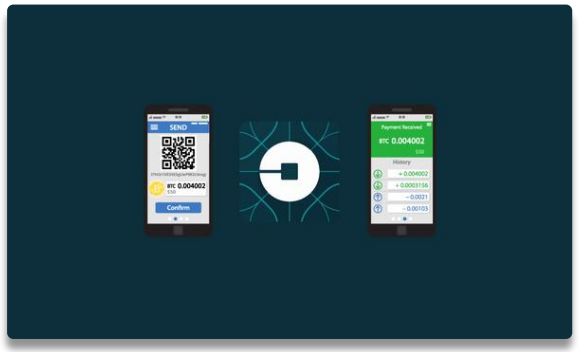
- **Extensive Testing**
- Including stress tests to evaluate scalability under varying loads, ensuring the blockchain network's robustness and efficiency.
- **Smart Contract Integration**
- Exploring the integration of smart contracts to enhance operational efficiency and automation within the Bride platform.
- **Ongoing Collaboration**
- Maintaining an active collaboration with blockchain developers and researchers to stay updated on advancements in the field.

User Experience Enhancement | 07

Enhancements for user experience:

User Onboarding

User-friendly onboarding with intuitive instructions for blockchain features



Transaction Process

Transparent transaction process with added details for rider and driver verification.

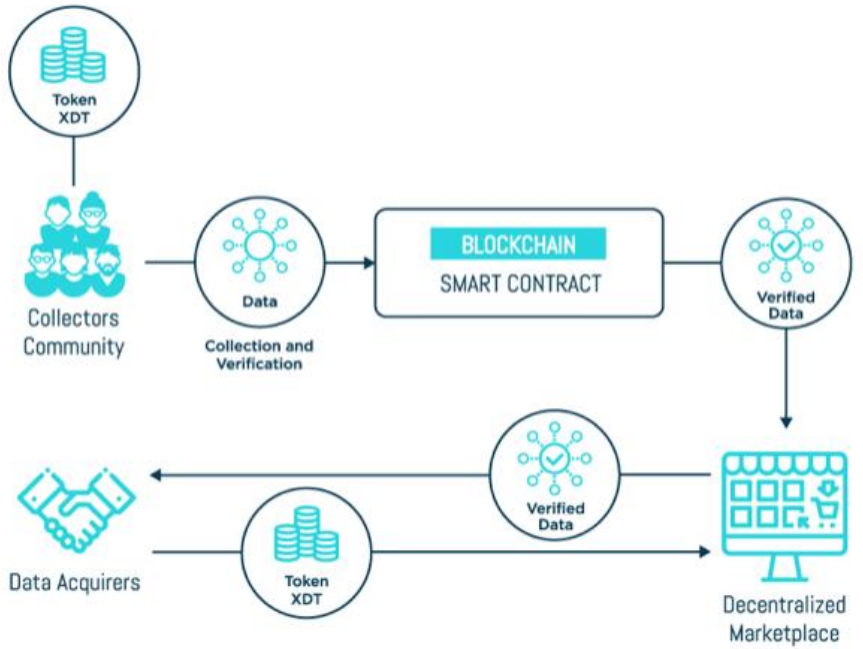
Mobile App Integration

Blockchain features integrated into the Bride app, simplifying user access and interactions.

Security and Privacy Considerations | 08

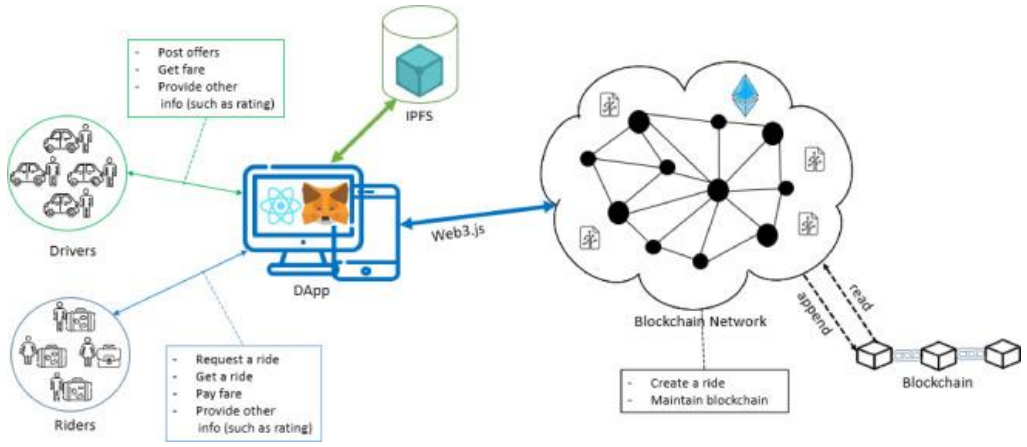
Focused security and privacy considerations

- **Consensus Mechanism**
- Avalanche's Byzantine fault tolerance ensures data integrity and protects against malicious nodes
- **Private Transactions**
- Cryptographic techniques for private transactions, balancing privacy and transparency.
- **Identity Management**
- Secure identity management empowers users while bolstering security.



Regulatory and Legal Implications | 09

Navigating legal and regulatory landscapes



- **Data Protection**
- Compliance with global data protection regulations, with user consent at the forefront.
- **Smart Contract Legality**
- Legal validation of smart contract enforceability
- **Cross-Border Transactions**
- Addressing complexities of cross-border transactions within regulatory frameworks.

Challenges and Mitigations | 10

Addressing challenges effectively:

• Scalability: Sharing techniques and optimized node configurations to manage increased transaction volume.

• User Adoption: User education and a user-friendly interface to encourage blockchain adoption.

• Integration Complexity: Gradual integration plan prioritizes payment processing and reduces disruption.

Collaboration and Partnerships | 11

Stakeholder engagement and collaborations:

• Blockchain Expertise: Leveraging insights from blockchain experts for optimal integration.

• Regulators and Policy Makers: Regular dialogue with regulators ensures compliance and support.

Business Model Innovation | 12

Innovating the business model:

• Tokenization Economy: Exploring the introduction of utility tokens for payments, rewards, and ecosystem participation.

• Platform Extensions: Enabling third-party developers to build blockchain-based applications for an enhanced Bride experience.

Ethical and Societal Impact | 13

Ethical and societal implications:

• Environmental Impact: Addressing energy consumption concerns through efficient blockchain network design.

• Equity and Inclusion: Ensuring inclusivity for marginalized communities through blockchain-enabled features.

Continuous Improvement and Monitoring | 14

Continuous enhancement and monitoring:

• Iterative Development: Continuously refining the integration based on user feedback and technological advancements.

• Performance Metrics: Establishing KPIs to gauge the success of the integration in terms of speed, accuracy, and user satisfaction.

Stakeholder Engagement | 15

Engaging stakeholders effectively:

• User Education: Launching a comprehensive educational campaign to communicate blockchain benefits to users.

• Feedback Mechanisms: Implementing feedback loops to gather user input for ongoing refinement.

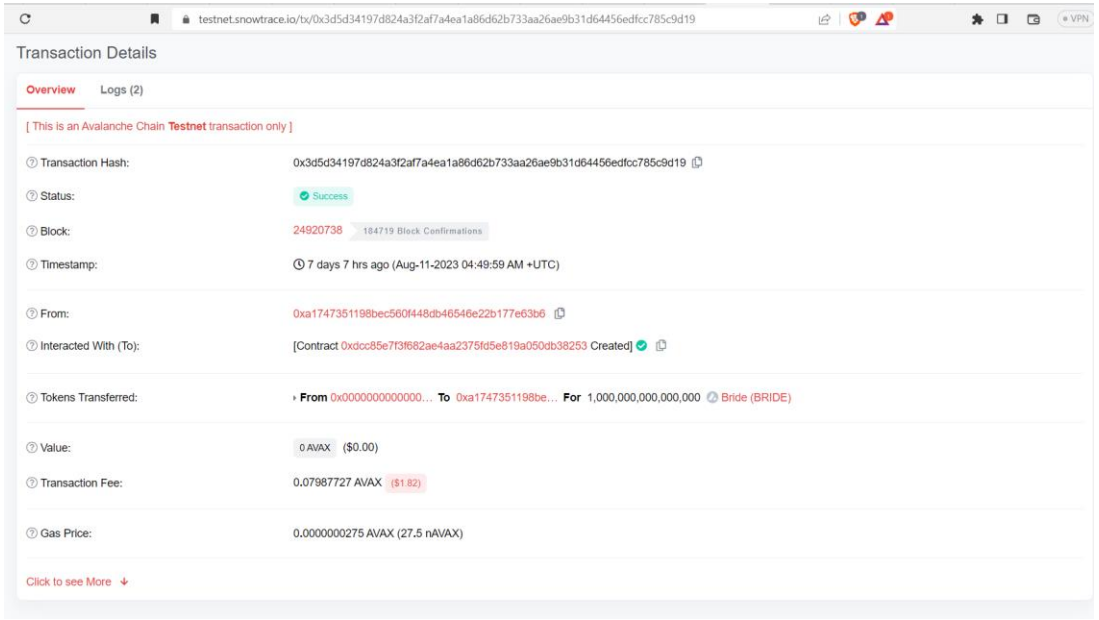
Conclusion | 16

- The Bride capstone project has successfully unveiled a comprehensive blueprint for integrating the Bride Avalanche blockchain protocol into the Bride platform. Through meticulous technical implementation, security considerations, legal compliance, and user experience enhancement, the project not only addresses challenges but also harnesses the inherent promise of blockchain technology.
- The findings underscore the potential of the Avalanche consensus protocol in enhancing transaction speeds, data integrity, and privacy within the ride-hailing ecosystem. As we continue on the journey to realize the full potential of blockchain technology in transportation, further research, testing, and collaboration with blockchain experts will be pivotal.

Appendix | 17

Project on snowtrace

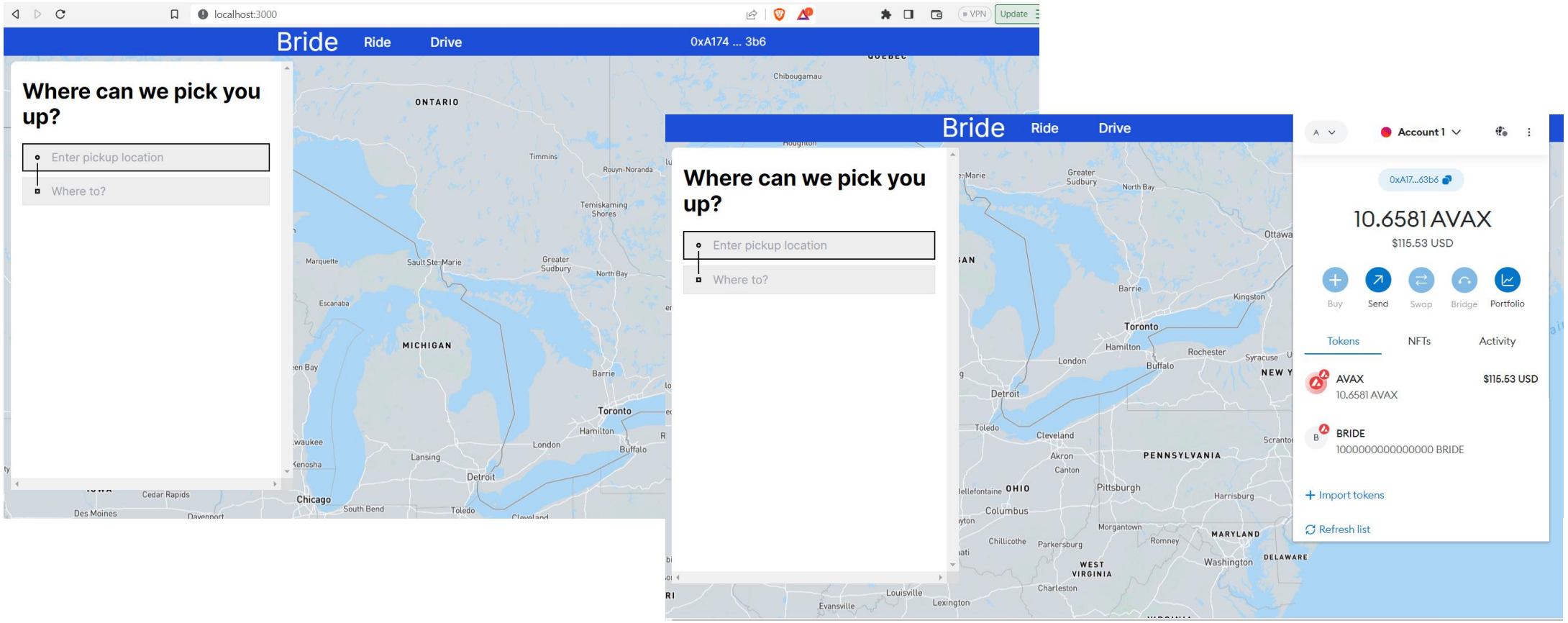
https://testnet.snowtrace.io/tx/0x3d5d34197d824a3f2af7a4ea1a86d62b733aa26ae9b31d64456edfcc785c9d19



DEMO BRIDE

Transaction of our project on snowtrace

https://testnet.snowtrace.io/address/0xA1747351198Bec560f448DB46546E22B177e63b6



Bride coin smart contract| 18

1.

```
// SPDX-License-Identifier: MIT LICENSE

import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";
import "@openzeppelin/contracts/access/AccessControl.sol";

pragma solidity ^0.8.7;

contract Bride is ReentrancyGuard, Ownable, AccessControl {

    uint256 public currentEntry; // This variable counts the amount of users registered.
    // currentEntry++ , this increments the integer by 1.

    struct userAccount {
        string accountCid;
        address userId;
        address payWallet;
        uint256 userNum;
    }
```

Bride coin smart contract | 18

2.

```
mapping(address => userAccount) public _account;

struct userNumber {
    address entryWallet;
}

mapping(uint256 => userNumber) public _entry;

struct userPicture {
    string pictureCid;
}

mapping(address => userPicture) public _picture;

struct erc20Pay {
    uint256 lastPaid;
}

mapping(address => erc20Pay) public _ercPay;

bytes32 public constant UPDATER_ROLE = keccak256("UPDATER_ROLE");

constructor() {
    _setupRole(DEFAULT_ADMIN_ROLE, _msgSender());
    _setupRole(UPDATER_ROLE, _msgSender());
}
```

Bride coin smart contract | 18

3.

```
function createProfile(string memory newCid, address wallet, address payErcWallet)
external
nonReentrant
{
 require(hasRole(UPDATER_ROLE, _msgSender()), "You must have updater role to run");
 currentEntry++;
 _account[wallet] = userAccount({
    accountCid: newCid,
    userId: wallet,
    userNum: currentEntry,
    payWallet: payErcWallet
    });
_entry[currentEntry] = userNumber({
     entryWallet: wallet
    });

}
```

4.

```
function updateProfile(string memory newCid, address wallet)
external
nonReentrant
{
  require(hasRole(UPDATER_ROLE, _msgSender()), "You must have updater role to run");
  address ercWallet = _account[msg.sender].payWallet;
  uint256 usernumber = _account[msg.sender].userNum;
  _account[wallet] = userAccount({
    accountCid: newCid,
    userId: wallet,
    payWallet: ercWallet,
    userNum: usernumber
    });
}
```

Bride coin smart contract | 18

5.

```
function updatePicture(string memory newCid, address wallet)
external
nonReentrant
{
  require(hasRole(UPDATER_ROLE, _msgSender()), "You must have updater role to run");
  _picture[wallet] = userPicture({
    pictureCid: newCid
   });
}

function recordPay(uint256 lastPay, address wallet)
external
nonReentrant
{
  require(hasRole(UPDATER_ROLE, _msgSender()), "You must have updater role to run");
  _ercPay[wallet] = erc20Pay({
    lastPaid: lastPay
    });
}
```

6.

```
function deleteProfile()
external
nonReentrant
{
  require(_account[msg.sender].userId == msg.sender, "Account not found");
  uint256 previousId = _account[msg.sender].userNum;
  delete _entry[previousId];
  delete _account[msg.sender];
  delete _picture[msg.sender];
  delete _ercPay[msg.sender];
}

function confirmUser() external view returns (address){
    address userWallet = _account[msg.sender].userId;
    return userWallet;
}

}
```

ACKNOWLEDGMENTS | 19

We extend our gratitude to our mentors, collaborators, and industry experts who provided invaluable guidance throughout the project. Their insights and expertise have been pivotal in shaping the project's direction and outcomes.

References | 20

- Buterin, V. (2013). Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform. Retrieved from https://ethereum.org/whitepaper/
- Team Rocket. (2021). Avalanche Consensus: The Next Big Breakthrough in Blockchain. Retrieved from https://cointelegraph.com/bitcoin-for-beginners/what-are-cryptocurrencies
- European Union Agency for Cybersecurity. (2020). Blockchain Security Guidance. Retrieved from https://www.enisa.europa.eu/publications/blockchain-security-guidance
- World Economic Forum. (2020). Blockchain Beyond the Hype: A Practical Framework for Business Leaders. Retrieved from https://www.weforum.org/whitepapers/blockchain-beyond-the-hype-a-practical-framework-for-business-leaders
- United Nations. (2019). Blockchain for Sustainable Development. Retrieved from https://www.un.org/development/desa/dpad/wp-content/uploads/sites/45/WP_317_2019.pdf
- Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf
- California Consumer Privacy Act (CCPA). (2018). California Civil Code. Retrieved from https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?lawCode=CIV&division=3.&title=1.81.5.&part=4.&chapter=&article=
- Zhang, Y., Jiang, X., Huang, Y., Zhao, J., & Liu, Y. (2018). Privacy-Preserving and Efficient Smart Contract Evaluation on Consortium Blockchain. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 1811-1828). ACM. European Central Bank. (2020). Report on a Digital Euro. Retrieved from https://www.ecb.europa.eu/pub/pdf/other/Report_on_a_digital_euro~4d726a0d01.en.pdf
- Bride Technologies Inc. (2023). Bride Platform Overview. Retrieved from https://www.Bride.com/us/en/platform/
- Lyft Inc. (2023). How Lyft Works. Retrieved from https://www.lyft.com/rider/how-it-works