PERFECTUS

WSO2 TOKEN & DAPP USE CASES & ARCHITECTURE

PREPARED FOR:
WSO2

PREPARED BY:
Perfectus (PVT) LTD

TABLE OF CONTENT

EXECUTIVE SUMMARY

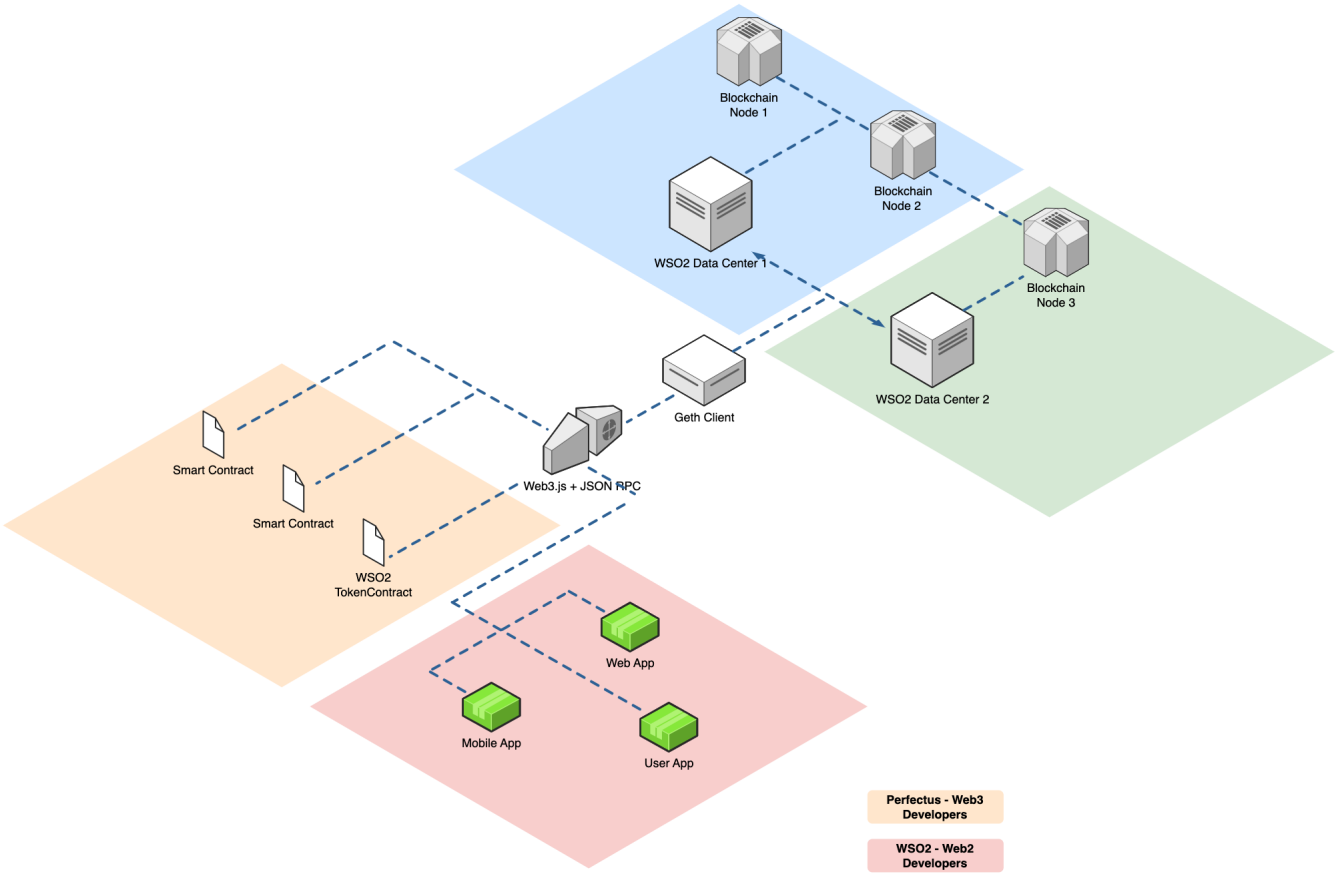
Our web3 development team, based in Sri Lanka, Ukraine, and the UK, is proud to announce the development of a new internal token ecosystem for WSO2, a leading software development company. With over 500 successful projects completed for clients worldwide and two years of experience in the web3 space, we are confident in our ability to deliver a top-notch solution for WSO2.

The internal token, referred to as the WSO2 token, will be a crucial component of this ecosystem, providing a digital representation of the contributions and achievements of employees at WSO2. The WSO2 token ecosystem, in combination with the Gamefi elements, will improve efficiency, transparency, and decentralization in the recognition and reward process for WSO2 employees. The WSO2 token can be exchanged for tangible rewards such as bonuses, promotions, and other benefits for the whole token ecosystem combined with activities.

By introducing an internal token ecosystem and Dapp activities, WSO2 aims to improve employee motivation and engagement, boost productivity, and create a more positive work culture. The token ecosystem and Dapp will also provide valuable data and insights into employee performance, allowing for data-driven decision-making and continuous improvement.

As a leading web3 development team, we are committed to delivering a secure, transparent, and user-friendly internal token ecosystem and Dapp for WSO2. Our expertise and experience in the web3 space, combined with our commitment to delivering high-quality solutions, make us the ideal partner for WSO2 in this exciting project.

TECHNICAL ARCHITECTURE



WSO2 TOKEN IN PRIVATE EVM

The Blockchain network will consist of nodes, which will be maintained by the company. Each node will have a copy of the Blockchain ledger, which contains all the transaction records of the rewards and bonuses that are distributed to employees.

To interact with the Blockchain network, employees will use a user interface or a mobile application. This will allow them to view their rewards, bonuses, and token balances and participate in the reward program by submitting their claims for eligible activities such as carpooling, completing tasks, and so on.

To incentivize and encourage participation, the company will issue its own digital token, which employees can earn and use for different purposes. The tokens will be distributed based on the employee's performance, level of participation, and contributions to the company.

To ensure the security and integrity of the network, we will implement a consensus mechanism that will require the approval of multiple nodes before any transaction can be recorded on the Blockchain. We will also use encryption and other security measures to protect the privacy and confidentiality of employee data.

In terms of the different apps that employees can build within the company, these will be based on smart contracts, which are self-executing contracts that are stored on the Blockchain. Employees can use these smart contracts to automate different processes, such as submitting expense claims or managing supply chain logistics. In return, they will be rewarded with tokens for their contributions.

Overall, this architecture will create a transparent and secure reward program that encourages positive behavior and contributions from employees. It will also enable the company to automate and streamline different processes, resulting in increased efficiency and productivity.

TECHNICAL ARCHITECTURE

ETHEREUM CLIENT

The Ethereum client software will allow you to interact with the Ethereum network, including deploying and executing smart contracts, managing accounts, and transferring tokens.

SMART CONTRACTS

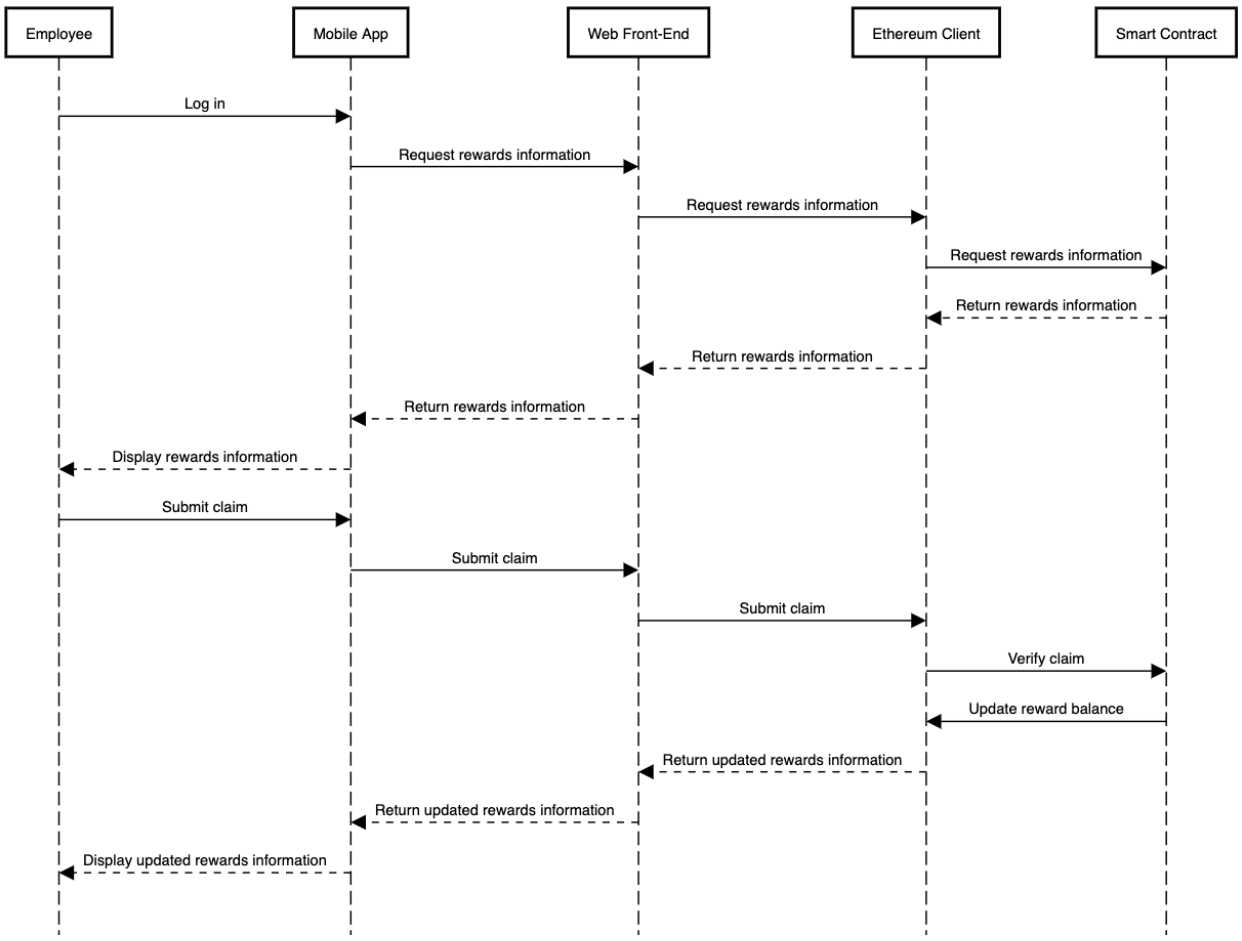
The employee reward program will be implemented as a set of smart contracts, which are self-executing code that runs on the Ethereum network. The smart contracts will define the rules and logic of the reward program, including how tokens are earned and distributed to employees.

TOKEN

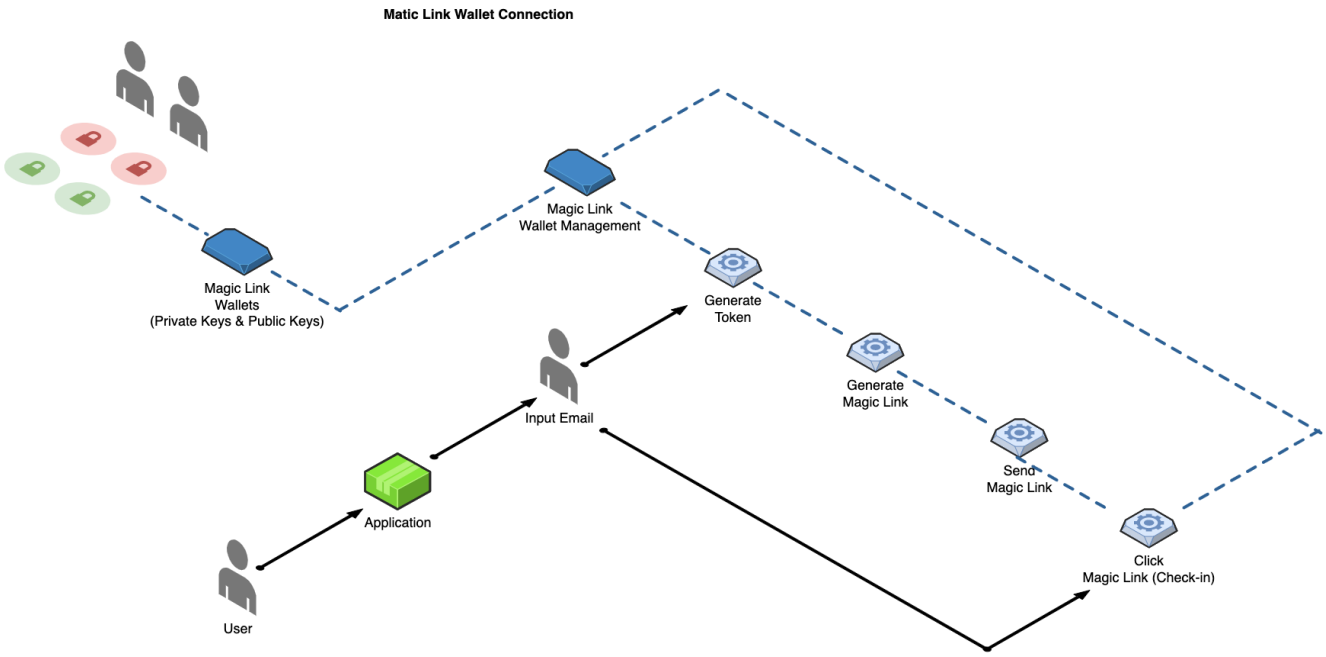
WSO2 will issue its own digital token, which employees can earn and use for different purposes within the company. The token will be distributed based on the employee's performance, level of participation, and contributions to the company.

USER INTERFACE

Employees will interact with the Ethereum network and the reward program using a user interface, which could be a web application, a mobile application, or both. The user interface will allow employees to view their rewards, bonuses, and token balances and participate in the reward program by submitting their claims for eligible activities.


Employee
Mobile App
Web Front-End
Ethereum Client
Smart Contract
Log in
Request rewards information
Request rewards information
Request rewards information
Return rewards information
Return rewards information
Return rewards information
Display rewards information
Submit claim
Submit claim
Submit claim
Verify claim
Update reward balance
Return updated rewards information
Return updated rewards information
Display updated rewards information

WALLET CONNECTION MECHANISM



Magic link connectivity is a secure and convenient way to connect private and public keys with a user's account without the need for users to manually enter and manage their keys.

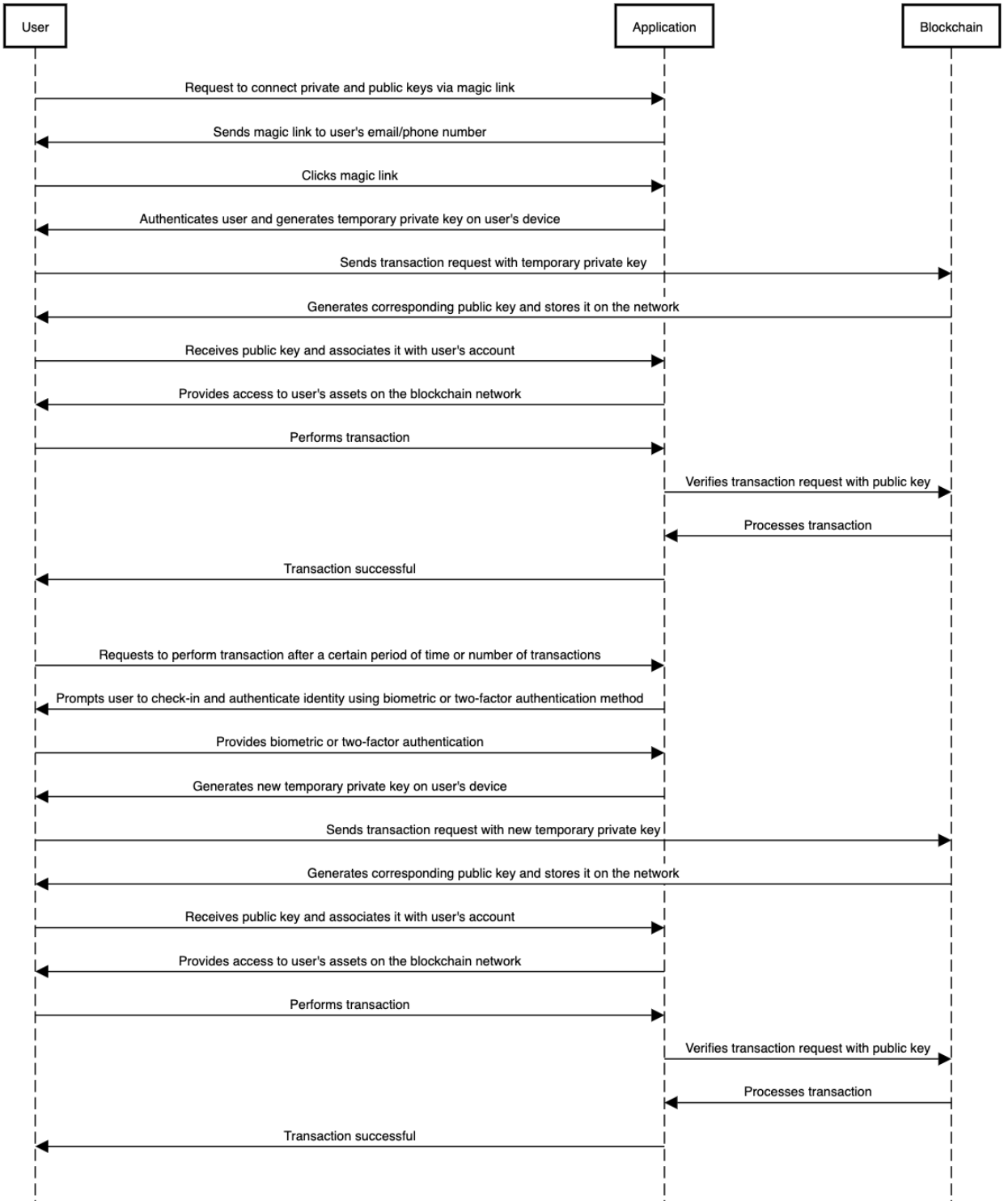
In this approach, a magic link is sent to the user's email or phone number, which, when clicked, authenticates the user and generates a unique, temporary private key that is securely stored on the user's device. This temporary key is used to sign transactions on the blockchain network, and a corresponding public key is generated and stored on the network.

The user's account is associated with the public key, allowing them to securely access and manage their assets on the blockchain network without exposing their private key.

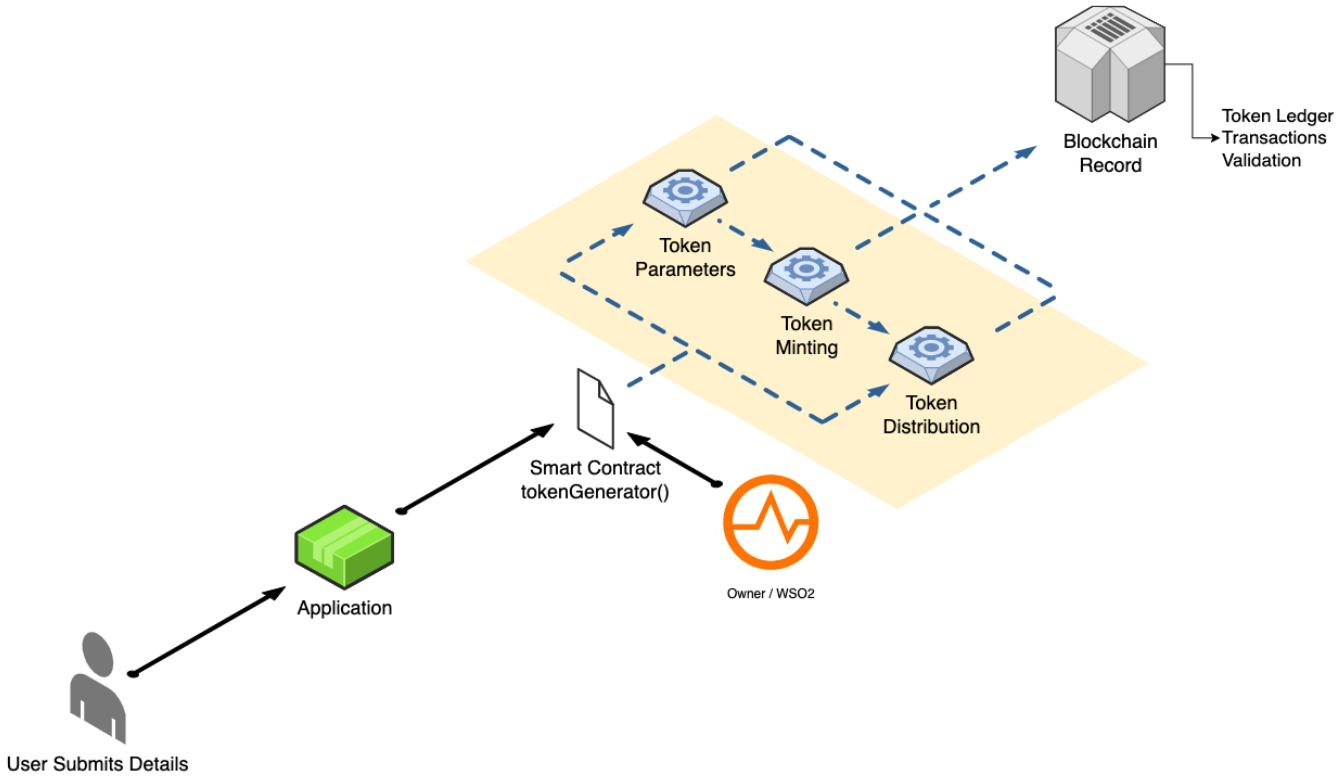
Additionally, check-in functionality can be used to enhance the security of this approach. For instance, after a certain period of time or number of transactions, the user may be required to check in and authenticate their identity using a biometric or two-factor authentication method.

This adds an extra layer of security to ensure that only authorized users can access and manage their assets on the blockchain network.

Overall, magic link connectivity with check-in functionality provides a secure and user-friendly way to manage private and public keys, enabling users to easily access and manage their assets on the blockchain network while maintaining a high level of security.

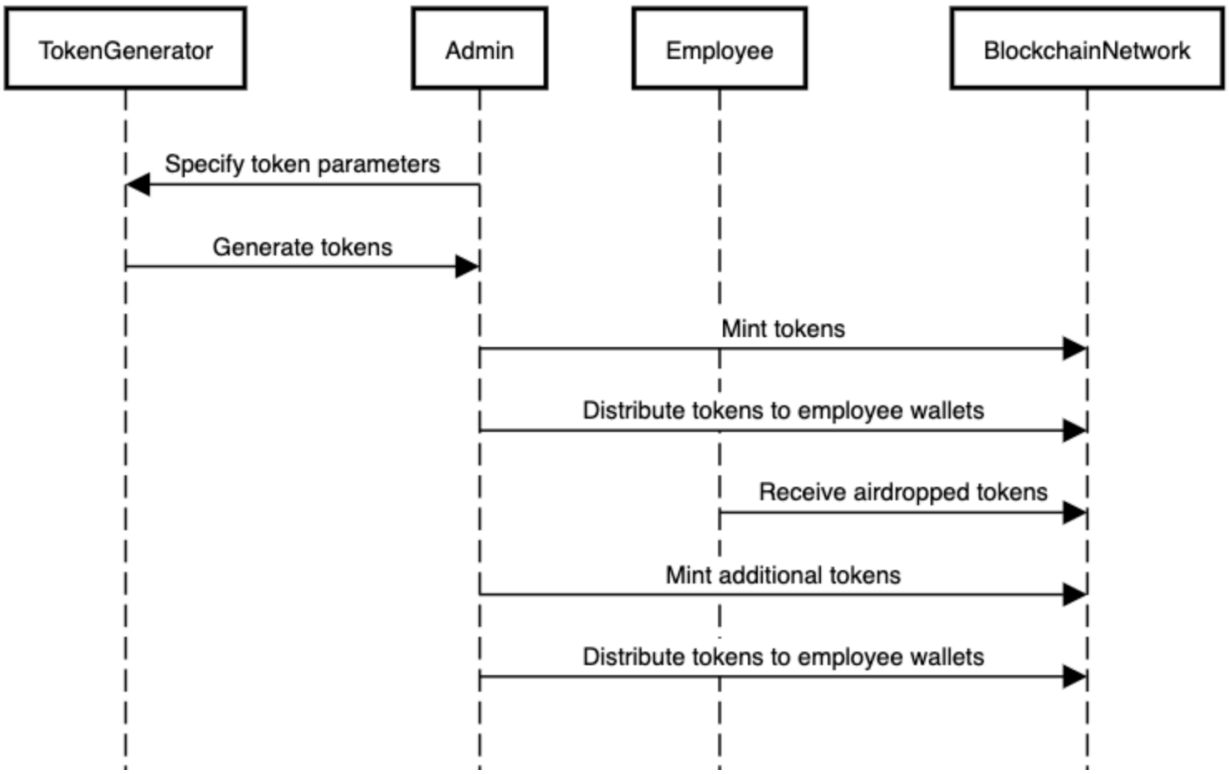


TOKEN DISTRIBUTION

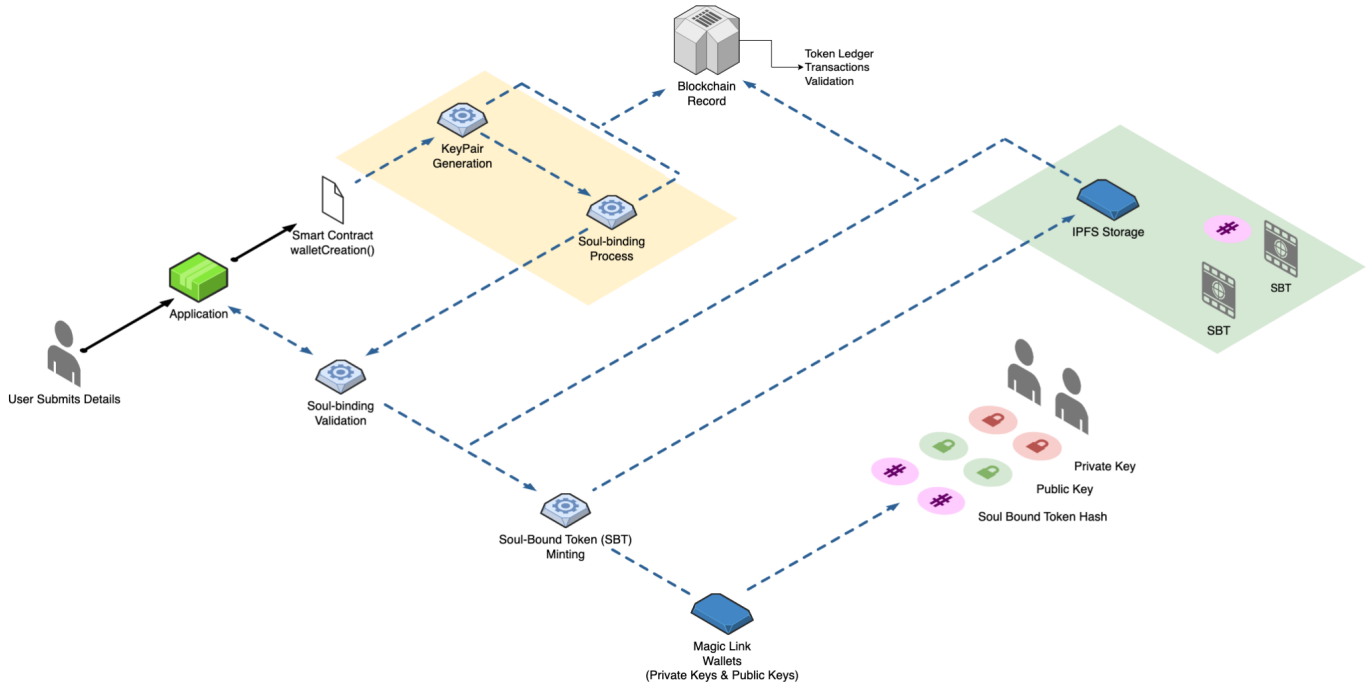


- X number of tokens will be minted as per what we decide in the initial stage.
- X number of tokens will be airdropped to all employee wallets upon the employee onboarding process.
- The main token will be inflationary and the owner with admin rights will be able to mint as many tokens in the future.
- Admin can distribute tokens anytime to the user wallets by calling smart contract functions.

SEQUENCE DIAGRAM



SOUL BOUND TOKEN & NFT GENERATION



SOUL BOUND TOKEN MINT AND TOKEN GENERATION

1. The token generator creates tokens according to the specified parameters, such as the total supply, token name, and symbol. The generator also initiates the soul-binding process, which connects the token with the employeer's soul.

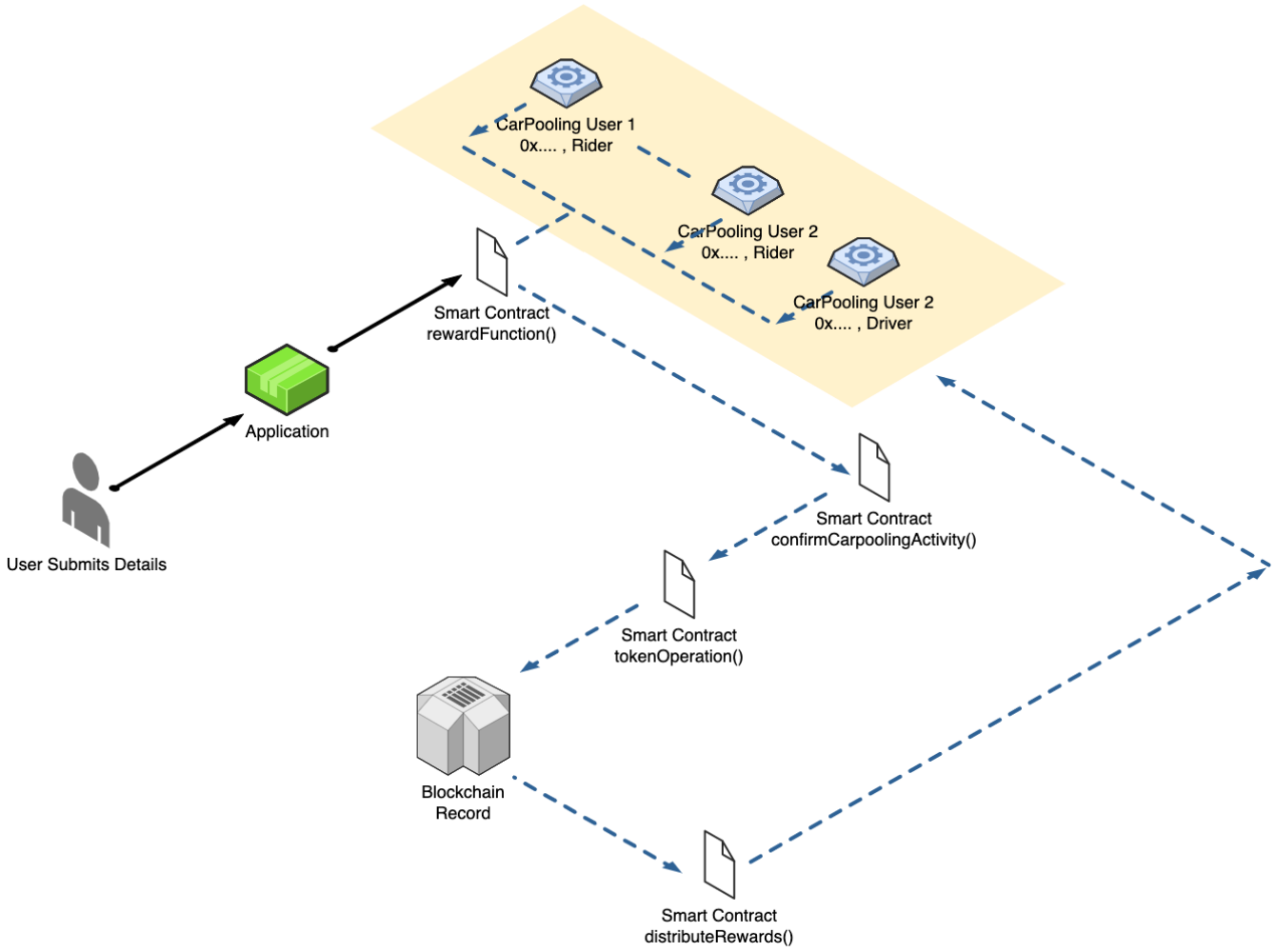
2. Once the tokens are generated, the owner creates a wallet by generating a key pair, which consists of a public key and a private key. The soul-binding process is then performed, which verifies the employeer's identity and links their soul to the token.

3. The wallet is stored on the blockchain network, and the owner's public key is used to represent their wallet address. The key pair is kept secret, and only the owner can access their wallet and perform transactions.

4. Transactions involving soul-bound tokens are validated by the blockchain network, which ensures that the transaction is authorized and that the employeer's soul is verified. The token ledger records all transactions involving the token, including transfers, purchases, and rewards.

Overall, this architecture provides a secure and efficient way to handle initial token distribution and wallet management with soul-bound tokens. The soul-binding process ensures that the token is tied to the owner's soul, providing a high level of security and preventing token theft or fraudulent transactions. The use of a blockchain network ensures transparency and immutability, providing a tamper-proof record of all token transactions.

REWARDS FOR CARPOOLING EMPLOYEES SCENARIO



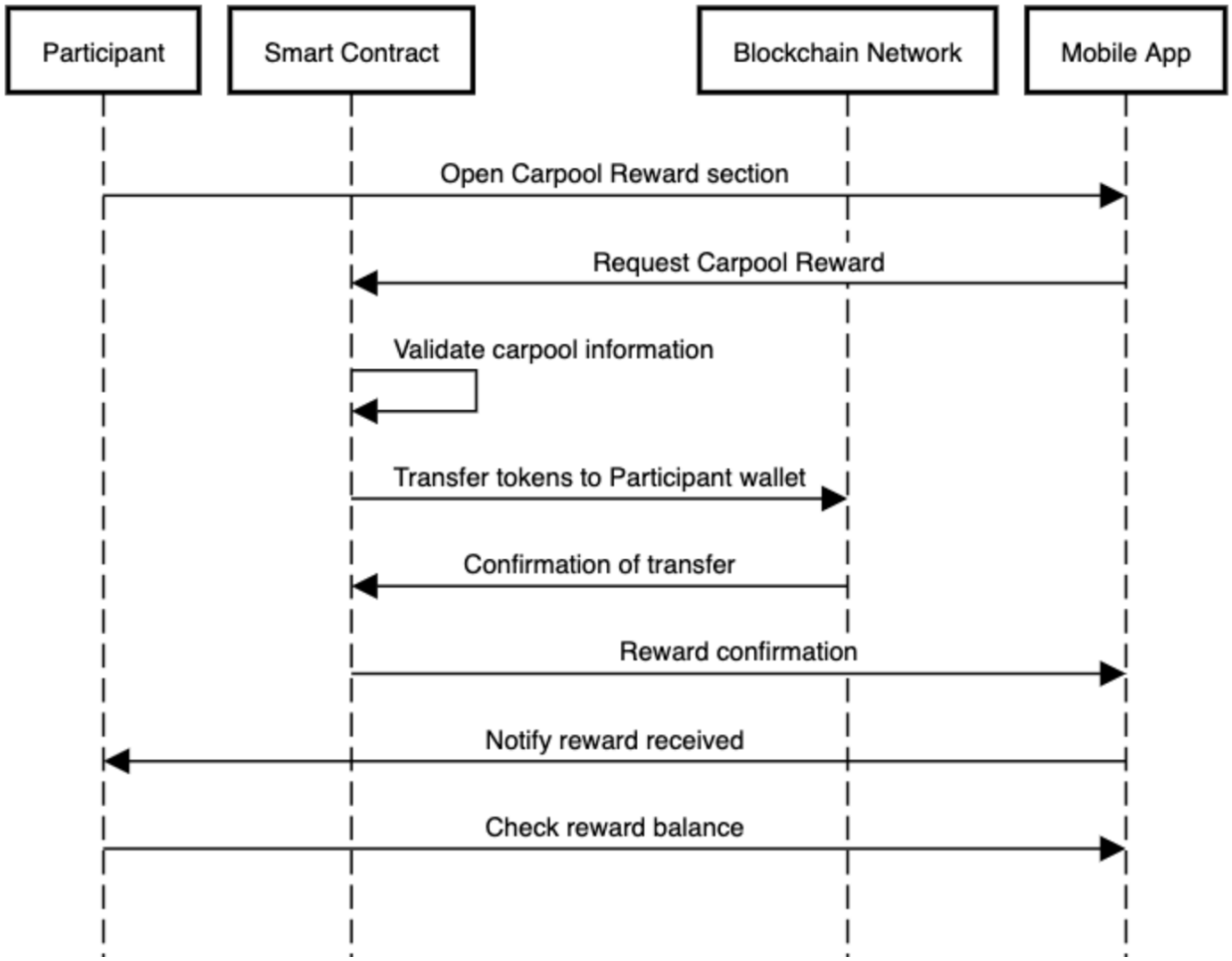
Carpool participants claim their rewards by submitting a request to the smart contract. This request includes information such as the date, time, and participants of the carpool activity.

The carpool activity is confirmed by the smart contract, which checks that the information provided by the participant is valid and meets the carpooling requirements set by the company.

If the carpooling activity is confirmed, the smart contract sends a token reward to the participant's wallet on the blockchain network.

Participants can check their token balances and transfer tokens to other participants or use them for other purposes.

This architecture provides a simple and efficient way to reward carpooling participants with tokens, promoting sustainable and environmentally friendly practices within the company. The use of a smart contract ensures that the reward system is transparent, secure, and automated, reducing administrative overhead and making it easy for participants to claim their rewards.



DEVELOPMENT MILESTONE

MILESTONE 01

- Signing the agreements
- Technical analysis of wso2 technologies and selection of the tech stack
- Scoping and documentation
- Laying out the overall project scope with the WSO2 team and signing off.

MILESTONE 02

- Private EVM blockchain set-up
- Wallet connect part (user onboarding process: Test Accounts)
- Soul-bound token smart contract creation.
- User profile layer development with wallet, soul-bound tokens, and user profile view.
- Testing and QA.

MILESTONE 03

- Development of WSO2 token smart contract (Testnet)
- Admin panel frontend development for the smart contract functions.
- Smart contract development & integration with driver pooling frontend.
- Testing and QA.
- Deployment to the production

MILESTONE 04

- Smart contract development & integration with Food serving process
- Dynamic event creation process development. (Main smart contract, smart contract pool creation, user-level frontend, admin-level data update and configurations)
- Testing and QA.
- Deployment to the production