



ALcoin Technical Documentation

Version 1.0

Date: April 12, 2025

Author: ALS sp. z o.o.

1. Introduction

ALcoin (ALC) is an innovative hybrid BEP-20 token built on the Binance Smart Chain, combining features of:

- investment token (participation in a real hotel project),
- utility token (discounts and services),
- loyalty token (staking, dividends).

The token was created as part of the tokenization project of the Wrzos Resort & Wellness hotel in Węgierska Góra.

2. Token Technical Data

Name: ALcoin

Symbol: ALC

Type: BEP-20 (Binance Smart Chain)

Total Supply: 25,000,000 ALC

Decimals: 18

Contract Address: [contract address after deployment]

Upgradeable: Yes (UUPS Proxy)

Solidity: 0.8.22

Framework: Hardhat + OpenZeppelin Upgradeable

RPC Provider: Infura (Sepolia/BSC)

3. Contract Functionalities

- 1) Upgradeability via UUPS Proxy
- 2) Whitelist for investors
- 3) Token purchase in presale
- 4) Ability to extend presale end date
- 5) Dividend up to 10% annually (paid every 6 months)
- 6) Staking with 3% annual reward
- 7) Token buyback after minimum 6 months
- 8) Discounts: 5% for accommodation, 10% in hotel restaurant
- 9) Price set in USD (0.28 USD) and converted to ETH
- 10) Token burning function
- 11) Purchase limit: min. 10,000 ALC

ALS Sp. z o.o.

NIP 625-21-21-456

Ul. Filarskiego 39, 47-330 Zdzeszowice



4. Public Methods (Functions)

- 1) buyTokens(uint256) - Purchase tokens through whitelist
- 2) addToWhitelist(address) - Add user to whitelist
- 3) stakeTokens(uint256) - Start staking
- 4) unstakeTokens() - End staking
- 5) claimDividends() - Claim available dividends
- 6) buyBackTokens(uint256) - Token buyback after 6 months
- 7) getTokenPriceInETH() - Convert ALC price to ETH
- 8) setPresaleEnd(uint256) - Set presale end date
- 9) setHotelStaff(address) - Set hotel staff address
- 10) burn(uint256) - Burn tokens
- 11) withdrawETH() - Withdraw funds

5. Security and Audit

- 1) ReentrancyGuardUpgradeable protects against reentrancy
- 2) OwnableUpgradeable restricts access to owner only
- 3) Roles: owner, staff, whitelisted investors
- 4) Planned external audit

6. External Integrations

- 1) Chainlink ETH/USD Price Feed
- 2) MetaMask (user interface)
- 3) Hardhat (deployment, testing)
- 4) Infura (RPC provider)

7. Deployment Instructions

- 1) .env Configuration:

```
SEPOLIA_RPC_URL=https://sepolia.infura.io/v3/YOUR_PROJECT_ID  
PRIVATE_KEY=YOUR_PRIVATE_KEY  
ETHERSCAN_API_KEY=YOUR_ETHERSCAN_KEY
```

- 2) Commands:

```
npx hardhat compile  
npx hardhat run scripts/deploy.js --network sepolia  
npx hardhat verify --network sepolia <IMPLEMENTATION_ADDRESS>
```

- 3) Additional Scripts:

- a. addToWhitelist.js
- b. buyTokens.js
- c. setPresaleEnd.js
- d. stakeTokens.js
- e. buyBackTokens.js

ALS Sp. z o.o.

NIP 625-21-21-456

Ul. Filarskiego 39, 47-330 Zdzeszowice



8. Usage Examples (User)

- 1) Write as Proxy > buyTokens (MetaMask)
- 2) stakeTokens > enter number of tokens
- 3) unstakeTokens > end staking
- 4) isWhitelisted(address) > true/false

9. License

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy.