VITALBlock security.

Blockchain Security | Smart Contract Audit | KYC Certification | SAFU | CEX Listing | Marketing

MADE IN CANADA

PRICEDICTION

AUDIT

SECURITY ASSESSMENT

19th November 2025

For

Making Blockchain, Defi And Web3 A Safer Place.

Smart Check SLITHER TRAIL OF BITS MythX

@VB_Audit @Vitalblock @VB_Audit www.vitalblock.org

CONTENTS

INTRODUCTION

Auditing Firm	**VITAL BLOCK SECURITY**
Client Firm	**PRICEDICTION**
Methodology	**Automated Analysis, Manual Code Review**
Language	**Rust**
Contract Address	8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS
Source Code Light	**Verified**
Centralization	**Active ownership**
Lifeguard	Always Informed. Always Ahead.
Blockchain	**SOLANA CHAIN**
Website	**https://pricediction.com/**
Twitter	**https://x.com/pricediction**
Medium	**https://medium.com/@pricediction.ai**
Docs	**https://pricediction.com/docs/pricediciton-whitepaper.pdf**
Prelim Report Date	**November 19TH 2025**
Final Report Date	**November 19TH 2025**

Verify the authenticity of this report on our GitHub Repo: https://www.github.com/vital-block

Document Properties

Client	**PRICEDICTION**
Title	Smart Contract Audit Report
Target	**PRICEDICTION**
Version	1.0
Author	Akhmetshin Marat
Auditors	Akhmetshin Marat, James BK, Ben Partrick , C. John
Reviewed by	Dima Meru
Approved by	Prince Mitchell
Classification	Public

Version Info

Version	Date	Author(s)	Description
1.0	November 19th , 2025	C. John	Final Release
1.0-AP	November 19th , 2025	C. John	Release Candidate

Contact

For more information about this document and its contents, please contact Vital Block Security Inc.

Name	Akhmetshin Marat
Phone	+1 (579) 817-7049
Email	info@vitalblock.org

In the following, we show the specific pull request and the commit hash value used in this audit.

- **PRICEDICTION** (PIRD331847)
- https://dexscreener.com/solana/9cm5bxy9bw4248nj4mhma9kjnz61utubhxr1jzsx3c3o (PLK877O54)

About Vital Block Security

Vital Block Security provides professional, thorough, fast, and easy-to-understand smart contract security audit. We do in-depth and penetrative static, manual, automated, and intelligent analysis of the smart contract. Some of our automated scans include tools like ConsenSys MythX, Mythril, Slither, Surya. We can audit custom smart contracts, DApps, NFTs, etc (including the service of smart contract auditing). We are reachable at Telegram (https://t.me/vitalblock), Twitter (http://twitter.com/Vb_Audit), or Email (info@vitalblock.org).

Table 1.2: Vulnerability Severity Classification

Impact \ Likelihood	*High*	*Medium*	*Low*
High	Critical	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Low

Methodology

To standardize the evaluation, we define the following terminology based on the OWASP Risk Rating Methodology.

- Likelihood represents how likely a particular vulnerability is to be uncovered and exploited in the wild;
- Impact measures the technical loss and business damage of a successful attack;
- Severity demonstrates the overall criticality of the risk.

SCOPE OF WORK

Vital Block was consulted by PRICEDICTION to conduct the smart contract audit of its. SOLIDITY (SOL) source code. The audit scope of work is strictly limited to the mentioned .Sol file only:

O.PRDN.RUST

External contracts and/or interfaces dependencies are not checked due to being out of scope.

Verify audited contract's contract address and deployed link below:

Public Contract Address	
8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS	
Contract Name	**PRICEDICTION**
Ticker	**$PRDN**
Total Supply	**1,000,000,000**

Executive Summary

This audit was conducted on the Solana program deployed at address 8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS.

Due to the nature of Solana's architecture (where on-chain programs are not directly readable as high-level source code), a complete audit requires access to the verified source code or a trusted binary along with symbol information and build artifacts.

AUDIT METHODOLOGY

Smart contract audits are conducted using a set of standards and procedures. Mutual collaboration is essential to performing an effective smart contract audit. Here's a brief overview of Vital Block Security auditing process and methodology:

CONNECT

- **The onboarding team gathers source codes, and specifications to make sure we understand the size, and scope of the smart contract audit.**

AUDIT

- **Automated analysis is performed to identify common contract vulnerabilities. We may use the following third-party frameworks and dependencies to perform the automated analysis:**
  - **Remix IDE Developer Tool**
  - **Open Zeppelin Code Analyzer**
  - **SWC Vulnerabilities Registry**
  - **DEX Dependencies, e.g., Pancakeswap, Uniswap**
- **Simulations are performed to identify centralized exploits causing contract and/or trade locks.**
- **A manual line-by-line analysis is performed to identify contract issues and centralized privileges. We may inspect below mentioned common contract vulnerabilities, and centralized exploits:**

Centralized Exploits	○ **Token Supply Manipulation** ○ **Access Control and Authorization** ○ **Assets Manipulation** ○ **Ownership Control** ○ **Liquidity Access** ○ **Stop and Pause Trading** ○ **Ownable Library Verification**

Common Contract Vulnerabilities

- Integer Overflow
- Lack of Arbitrary limits
- Incorrect Inheritance Order
- Typographical Errors
- Requirement Violation
- Gas Optimization
- Coding Style Violations
- Re-entrancy
- Third-Party Dependencies
- Potential Sandwich Attacks
- Irrelevant Codes
- Divide before multiply
- Conformance to Solidity Naming Guides
- Compiler Specific Warnings
- Language Specific Warnings

REPORT

- The auditing team provides a preliminary report specifying all the checks which have been performed and the findingsthereof.
- The client's development team reviews the report and makes amendments to the codes.
- The auditing team provides the final comprehensive report with open and unresolvedissues.

PUBLISH

- The client may use the audit report internally or disclose it publicly.

ℹ It is important to note that there is no pass or fail in the audit, it is recommended to view theaudit as an unbiased assessment of the safety of solidity codes.

Table 1.0 The Full Audit Checklist

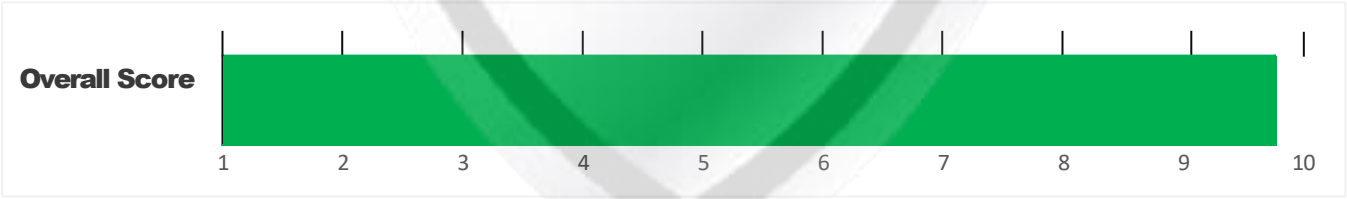
Category	Checklist Items
Basic Coding Bugs	Constructor Mismatch
	Ownership Takeover
	Redundant Fallback Function
	Overflows & Underflows
	Reentrancy
	Money-Giving Bug
	Blackhole
	Unauthorized Self-Destruct
	Revert DoS
	Unchecked External Call
	Gasless Send
	Send Instead Of Transfer
	Costly Loop
	(Unsafe) Use Of Untrusted Libraries
	(Unsafe) Use Of Predictable Variables
	Transaction Ordering Dependence
	Deprecated Uses
Semantic Consistency Checks	Semantic Consistency Checks
Advanced DeFi Scrutiny	Business Logics Review
	Functionality Checks
	Authentication Management
	Access Control & Authorization
	Oracle Security
	Digital Asset Escrow
	Kill-Switch Mechanism
	Operation Trails & Event Generation
	ERC20 Idiosyncrasies Handling
	Frontend-Contract Integration
	Deployment Consistency
	Holistic Risk Management
Additional Recommendations	Avoiding Use of Variadic Byte Array
	Using Fixed Compiler Version
	Making Visibility Level Explicit
	Making Type Inference Explicit
	Adhering To Function Declaration Strictly
	Following Other Best Practices

EXECUTIVE SUMMARY

Vital Block Security has performed the automated and manual analysis of the PRICEDICTION code. The code was reviewed for common contract vulnerabilities and centralized exploits. Here's aquick audit summary:

Status	Critical !	Major "	Medium #	Minor $	Unknown %
Open	0	0	1	0	0
Acknowledged	0	0	0	2	0
Resolved	0	0	1	1	0
Noteworty OnlyOwner Privileges	Set Taxes and Ratios, Airdrop, Set Protection Settings, Set Reward Properties, Set Reflector Settings, Set Swap Settings, Set Pair and Router				

PRICEDICTION Smart contract has achieved the following score: 97.0



i Please note that smart contracts deployed on blockchains aren't resistant to exploits, vulnerabilities and/or hacks. Blockchain and cryptography assets utilize new and emerging technologies. These technologies present a high level of ongoing risks. For a detailed understanding of risk severity, source code vulnerability, and audit limitations, kindly review the audit report thoroughly.

i Please note that centralization privileges regardless of their inherited risk status - constitute an elevated impact on smart contract safety andsecurity.

RISK CATEGORIES

Smart contracts are generally designed to hold, approve, and transfer tokens. This makes them very tempting attack targets. A successful external attack may allow the external attacker to directly exploit. A successful centralization-related exploit may allow the privileged role to directly exploit. All risks which are identified in the audit report are categorized here for the reader to review:

Risk Type	Definition
Critical	These risks could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
Major	These risks are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to high-risk severity.
Medium	These risks should be fixed, as they carry an inherent risk of future exploits, and hacks which may or may not impact the smart contract execution. Low-risk re-entrancy-related vulnerabilities should be fixed to deter exploits.
Minor	These risks do not pose a considerable risk to the contract or those who interact with it. They are code-style violations and deviations from standard practices. They should be highlighted and fixed nonetheless.
Unknown	These risks pose uncertain severity to the contract or those who interact with it. They should be fixed immediately to mitigate the risk uncertainty.

All statuses which are identified in the audit report are categorized here for the reader to review:

Status Type	Definition
Open	Risks are open.
Acknowledged	Risks are acknowledged, but not fixed.
Resolved	Risks are acknowledged and fixed.

CENTRALIZED PRIVILEGES

Centralization risk is the most common cause of cryptography asset loss. When a smart contract has a privileged role, the risk related to centralization iselevated.

There are some well-intended reasons haveprivileged roles, such as:

- **Privileged roles can be granted the power to** pause()**the contract in case of an external attack.**
- **Privileged roles can use functions like**, include(), **and** exclude() **to add or remove wallets from fees, swap checks, and transaction limits. This is useful to run a presale and to list on an exchange.**

Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Lately, centralization-related losses are increasing in frequency and magnitude.

- **The client can lower centralization-related risks by implementing below mentioned practices:**
- **Privileged role's private key must be carefully secured to avoid any potential hack.**
- **Privileged role should be shared by multi-signature (multi-sig) wallets.**
- **Authorized privilege can be locked in a contract, user voting, or community DAO can be introduced to unlock the privilege.**
- **Renouncing the contract ownership, and privilegedroles.**
- **Remove functions with elevated centralization risk.**

Understand the project's initial asset distribution. Assets in the liquidity pair should be locked. Assets outside the liquidity pair should be locked with arelease schedule.

AUTOMATED ANALYSIS

Symbol	Definition
	**Function modifies state**
	**Function is payable**
	**Function is internal**
	**Function is private**
	Function is important

| **PRICEDICTION** | Interface | |||

| └ | totalSupply | External ❗ | |NO❗| |

| └ | decimals | External ❗ | |NO❗| |

| └ | symbol | External ❗ | |NO❗| |

| └ | name | External ❗ | |NO❗| |

| └ | getOwner | External ❗ | |NO❗| |

| └ | balanceOf | External ❗ | |NO❗ |

| └ | transfer | External ❗ | " |NO❗ |

| └ | allowance | External ❗ | |NO❗ |

| └ | approve | External ❗ | " |NO❗ |

| └ | transferFrom | External ❗ | " |NO❗|

||||||

| **IFactoryV2** | Interface | |||

| └ | getPair | External ❗ | |NO❗| |

| └ | createPair | External ❗ | " |NO❗| |

||||||

| **IV2Pair** | Interface | |||

| └ | factory | External ❗ | |NO❗| |

| └ | getReserves | External ❗ | |NO❗| |

| └ | sync | External ❗ | " |NO❗| |

|||||

| **IRouter01** | Interface | |||

| └ | factory | External ❗ | | |NO❗ |

| └ | SOL| External ❗ | | |NO❗ |

| └ | addLiquiditySOL| External ❗ | | # |NO❗ |

| └ | addLiquidity | External ❗ | " | |NO❗ |

| └ | swapExacSOLTokens | External ❗ | | # |NO❗ |

| └ | getAmountsOut | External ❗ | | |NO ❗ |

| └ | getAmountsIn | External ❗ | | |NO ❗ |

| **IRouter02** | Interface | IRouter01 |||

| └ | swapExactTokensForSOLSupportingFeeOnTransferTokens | External ❗ | " | |NO❗ |

| └ | swapExactSOLForTokensSupportingFeeOnTransferTokens | External ❗ | | # |NO❗ |

| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | " ! 🛑 |NO❗ |

| └ | swapExactTokensForTokens | External ❗ | " | |NO❗ |

|||||

| **Protections** | Interface | |||

| └ | checkUser | External ❗ | " ! 🛑 |NO❗ |

| └ | setLaunch | External ❗ | " | |NO❗ |

| └ | setLpPair | External ❗ | " | |NO❗ |

| └ | PRDN | External ❗ | "! 🛑 |NO❗ ! |

| └ | removeSniper | External ❗ |!" 🛑 |NO❗ |

|||||

| **Cashier** | Interface | |||

| └ | setRewardsProperties | External ❗ | " | |NO❗ |

| └ | tally | External ❗ ! " 🛑 |NO❗ |

| └ | load | External ❗ ! 💵 |NO❗ |

| └ | cashout | External ❗ | " ! 🛑 |NO❗ |

| └ | giveMeWelfarePlease | External ❗ | " ! 🛑 |NO❗ |

| └ | getTotalDistributed | External ❗ | ! |NO❗ |

| └ | getUserInfo | External ❗ | ! |NO❗ |

| └ | getUserRealizedRewards | External ❗ | ! |NO❗ |

| └ | getPendingRewards | External ❗ | | ❗ |NO❗|

| └ | initialize | External ❗ | " ❗ 🛑 |NO❗|

| └ | getCurrentReward | External ❗ | |NO❗|

||||||

| **SOL ** | Implementation | **SafeMath** |||

| └ | <Constructor> | Public ❗ | ❗ |NO❗|

| └ | transferOwner | External ❗ | " ❗ 🛑 | onlyOwner |

| └ | renounceOwnership | External ❗ | " ❗ 🛑 | NO ❗

| └ | setOperator | Public ❗ | " ❗ 🛑 |NO❗|

| └ | renounceOriginalDeployer | External ❗ | " 🛑 |NO❗|

| └ | <Receive SOL> | External ❗ | ❗ |NO❗|

| └ | totalSupply | External ❗ | ❗ |NO❗|

| └ | decimals | External ❗ | ❗ |NO❗|

| └ | symbol | External ❗ | ❗ |NO❗|

| └ | name | External ❗ | ❗ |NO❗|

| └ | getOwner | External ❗ | ❗ |NO❗|

| └ | balanceOf | Public ❗ | ❗ |NO❗|

| └ | allowance | External ❗ | ❗ |NO❗|

| └ | approve | External ❗ | " ❗ 🛑 |NO❗|

| └ | _approve | Internal $ | "🔒 🛑 ||

| └ | approveContractContingency | Public ❗ | " ❗ 🛑 | onlyOwner |

| └ | transfer | External ❗ | " ❗ 🛑 |NO❗|

| └ | transferFrom | External ❗ | " ❗ 🛑 |NO❗|

| └ | setNewRouter | External ❗ | " ❗ 🛑 | onlyOwner |

| └ | setLpPair | External ❗ | " ❗ 🛑 | onlyOwner |

| └ | setInitializers | External ❗ | " ❗ 🛑 | onlyOwner |

| └ | isExcludedFromFees | External ❗ | ❗ |NO❗|

| └ | isExcludedFromDividends | External ❗ | ❗ |NO❗|

| └ | isExcludedFromProtection | External ❗ | ❗ |NO❗|

| └ | setDividendExcluded | Public ❗ | " | onlyOwner |

| └ | setExcludedFromFees | Public ❗ | "❗ 🛑 | onlyOwner |

PRICEDICTION - 01 POSSIBLE OVERFLOW

Category	Severity	Location	Status
Status Mathematical Operations	Critical	.PRDN ()	Acknowledged

Description

In Source Code Unverified

The program at 8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS does not have its source code published or verified on any public repository or explorer. Without the source, it is impossible to verify logic correctness, access controls, or business rules.

Impact: Users cannot confirm whether the program contains malicious logic (e.g., backdoors, fund theft, rug-pull mechanisms)

Recommendation

* The developer should publish the source code under an open-source license.
* Submit the program to Solana's verified programs registry or use tools like anchor verify.
* Until then, treat the program as untrusted.

```
mint:
"63bpnCja1pGB2HSazkS8FAPAUkYgcXoDwYHfvZZveBot"
data:{
4 items
name:
"Pricedicti
on" symbol:
"PRDN"
```

PRICEDICTION - 02 POSSIBLE OVERFLOW

Category	Severity	Location	Status
Inconsistency	**Informational**	Constructor and manageFeeExclusion()	Acknowledged

Description

In **Unknown Instruction Set**

No public documentation or IDL (Interface Description Language) exists for this program. Users cannot determine valid instruction formats or expected accounts.

Impact: Risk of malformed transactions, unintended behavior, or exploitation via unexpected account combinations.

Recommendation

Publish an Anchor IDL or equivalent instruction specification.

Program Address: 8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS
Owner: BAGSB9TpGrZxQbEsrEznv5jXXdwyP6AXerN8aVRiAmcv → Upgradeable

OPTIMIZATIONS | $PRDN

ID	Title	Category	Status
ERR	**Logarithm Refinement Optimization**	Gas Optimization	Acknowledged
YUU	**Checks Can Be Performed Earlier**	Gas Optimization	Acknowledged
BGH	**Unnecessary Use Of SafeMath**	Gas Optimization	Acknowledged
JUP	**Struct Optimization**	Gas Optimization	Acknowledged
WEE	**Unused State Variable**	Gas Optimization	Acknowledged

General Detectors

Missing Zero Address Validation

Some functions in this contract may not appropriately check for zero addresses being used.

Attention Required

Consistent Rust Version

This contract uses a conventional or very New version of Sol dependency

Attention Required

- No compiler version inconsistencies found
- No unchecked call responses found
- No vulnerable self-destruct functions found
- No assertion vulnerabilities found
- No old Rust code found
- No external delegated calls found
- No external call dependency found
- No vulnerable authentication calls found
- No invalid character typos found
- No RTL characters found
- No dead code found
- No risky data allocation found
- No uninitialized state variables found
- No uninitialized storage variables found
- No vulnerable initialization functions found
- No risky data handling found
- No number accuracy bug found
- No out-of-range number vulnerability found
- No map data deletion vulnerabilities found
- No tautologies or contradictions found
- No faulty true/false values found
- No innacurate divisions found
- No redundant constructor calls found
- No vulnerable transfers found
- No vulnerable return values found
- No uninitialized local variables found
- No default function responses found
- No missing arithmetic events found
- No missing access control events found
- No redundant true/false comparisons found
- No state variables vulnerable through function calls found
- No buggy low-level calls found
- No expensive loops found
- No bad numeric notation practices found
- No missing constant declarations found
- No missing external function declarations found
- No vulnerable payable functions found
- No vulnerable message values found

Vulnerability Scan

REENTRANCY

No reentrancy risk found

Severity	Minor
Confidence Parameter	Certain

Vulnerability Description

NOTMINTABLE: No additional amount of BOT token can be Determined minted by a private wallet or contract.
(Which is normal for major contract utility options)

Scanning Line:


```
{
updateAuthority:
"BAGSB9TpGrZxQbEsrEznv5jXXdwyP6AXerN8aVRiAmcv"
mint:
"8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS"
data:{
4 items
name:
"Pricediction"
symbol:
"PRDN"
uri:
"https://ipfs.io/ipfs/QmVthpgGnXa8TgDaKWmGzcBEyX8QG3UdpcpepbTp
jDpwTx"
sellerFeeBasisPoints:
0
}
```

Auto Contract Scan

Basic Info

Token Contract Address	8b1o3X...BAGS
Total Supply	1B
Issue Platform	bags.fm

Risk Check

- Doesn't look like honeypot
- Owner can not tamper with balance
- Doesn't look like a proxy contract
- No blacklist
- Admin privileges abandoned
- Can not Mint
- lp has burned

Mechanism Introduction

Buy Tax	0%
Sell Tax	0%

Token Holders Info

Token Holders: 186

Top10 ratio(exclude blackhole)	44.1%
1.Fh...HLuM	192.86M (19.29%)
2.8m...WnEd	31.65M (3.17%)
3.2v...QzCc	31.27M (3.13%)
4.68...evpr	30.61M (3.06%)
5.DB...csg7	28.63M (2.86%)
6.Fu...avMc	28.35M (2.84%)
7.58...UMxe	26.55M (2.65%)
8.8u...TcpK	25.83M (2.58%)
9.3h...ash3	23.4M (2.34%)
10.2L...FsQm	21.82M (2.18%)

More Details

LP

LP Holders: 1 — Total Supply: 1

Percentage of LP locked	100%
1. Bonding Curve LP Lock	1 (100%)

More Details

Contract Creator Address:

8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS

Audited Files

$PRDN

Contracts Creator Hash:

HASH:
2pGFXhMViV5ztW59hYykufQyzs8XeWRX8cJJkHS1Zute
viR7ZbAMNTzc8yzQjodNfikipYvguvTvpjq5dgedL6FA

Contracts:

Contract Address:
PRDN:8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BAGS

MANUAL REVIEW

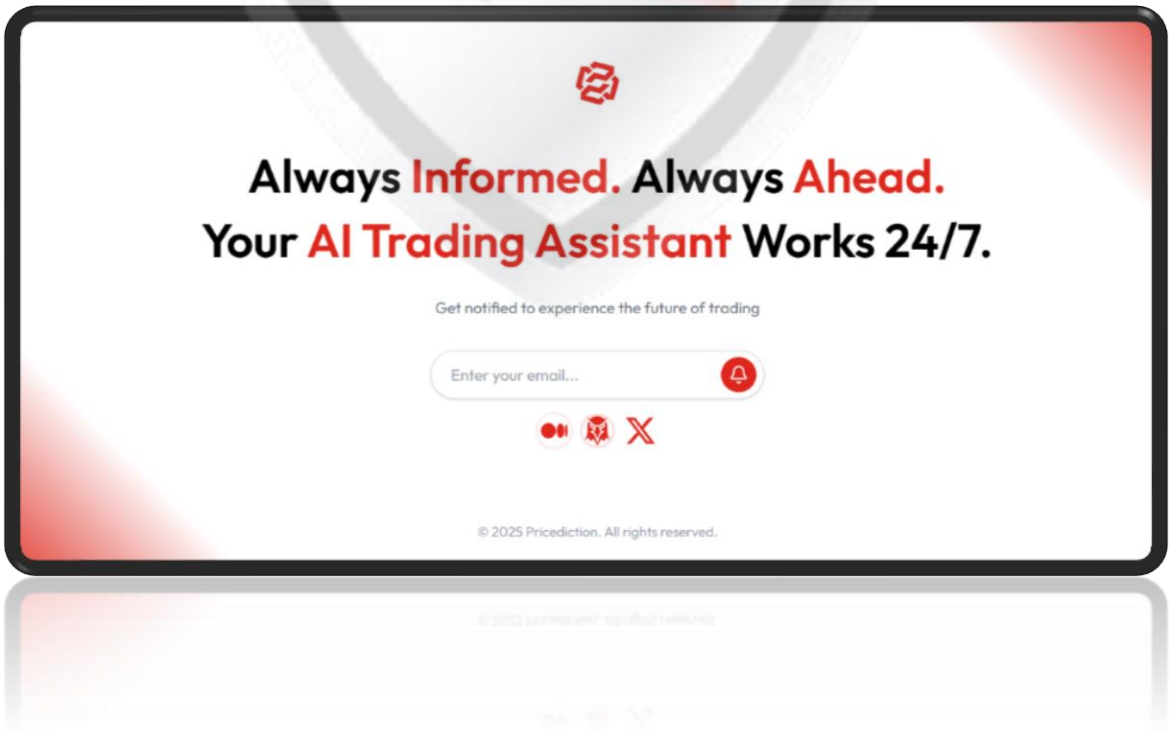
PRICEDICTION: Pricediction is not just another research tool. It is designed to embody the lessons of superforecasting and scale them with AI.

TOKEN NAME: PRICEDICTION
Ticker: $PRDN

Chain/Standard: SOLANA NETWORK

LAUNGUGE: RUST

The PRICEDICTION Platform Is Launching On the Solana Network



ISSUES CHECKING STATUS

VITALBlock security.

	Issue Description	Checking Status
1.	Compiler errors.	PASSED
2.	RaceConditionsand reentrancy. Cross-Function Race Conditions.	PASSED
3.	Possible Delay InData Delivery.	PASSED
4.	Oraclecalls.	PASSED
5.	Front Running.	PASSED
6.	RUSTDependency.	PASSED
7.	Integer OverflowAnd Underflow.	PASSED
8.	DoS with Revert.	PASSED
9.	Dos With Block GasLimit.	PASSED
10.	Methodsexecutionpermissions.	PASSED
11.	Economy Model of thecontract.	PASSED
12.	TheImpact Of ExchangeRateOnthe MoveLogic.	PASSED
13.	Privateusedata leaks.	PASSED
14.	MaliciousEvent log.	PASSED
15.	Scopingand Declarations.	PASSED
16.	Uninitialized storage pointers.	PASSED
17.	Arithmetic accuracy.	PASSED
18.	DesignLogic.	PASSED
19.	Cross-Function raceConditions	PASSED
20.	Save Upon Move contract Implementationand Usage.	PASSED
21.	Fallback FunctionSecurity	PASSED

AUDIT RESULT

PASSED

Identifier	Definition	Severity
CEN-02	Initial asset distribution	Minor

All of the initially minted assets are sent to the contract deployer when deploying the contract. This can be an issue as the deployer and/or contract owner can distribute tokens without consulting the community.


```
[1 item
0:{1 item
state:{8 items
key:{1 item
enumType:"metadataV1"
}
updateAuthority:"9hn2rhJ9Fepznmw6FKjTzoZPCQCD1qXHSXfqP3crcRQo"
mint:" 8b1o3XPCsEiWpcBKPq7fA3y1UDzBmDvupxt754v1BA "
data:{4 items
name:"PRICEDICTION"
symbol:"PRDN
uri:"https://gateway.pinata.cloud/ipfs/QmbuttgQ2MaMooCZVMWKb1gtnLfFmHyzo58RqE62YDzEbx"
sellerFeeBasisPoints:"0"
}
primarySaleHappened:false
isMutable:false
editionNonce:"255"
tokenStandard:{1 item
enumType:"fungible"
}
}
}
]
```


RECOMMENDATION

Project stakeholders should be consulted during the initial asset distributionprocess.

RECOMMENDATION

Deployer and/or contract owner private keys are secured carefully.

Please refer to PAGE-09 CENTRALIZEDPRIVILEGES for a detailed understanding.

ALLEVIATION

The PRICEDICTION project team understands the centralization risk. Some functions are provided privileged access to ensure a good runtime behavior in the project

CERTIFICATE BY VITAL BLOCK SECURITY

Identifier	Definition	Severity
COD-10	Third Party Dependencies	Minor

Smart contract is interacting with third party protocols e.g., Pancakeswap router, cashier contract, protections contract. The scope of the audit treats third party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised, and exploited. Moreover, upgrades in third parties can create severe impacts, e.g., increased transactional fees, deprecation of previous routers, etc.

RECOMMENDATION

Inspect and validate third party dependencies regularly, and mitigate severe impacts whenever necessary.

DISCLAIMERS

Vital Block provides the easy-to-understand audit of Solidity, Move and Raw source codes (commonly known as smart contracts).

The smart contract for this particular audit was analyzed for common contract vulnerabilities, and centralization exploits. This audit report makes no statements or warranties on the security of the code. This audit report does not provide any warranty or guarantee regarding the absolute bug-free nature of the smart contract analyzed, nor do they provide any indication of the client's business, business model or legal compliance. This audit report does not extend to the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. Cryptographic tokens are emergent technologies, they carry high levels of technical risks and uncertainty. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. This audit report could include false positives, false negatives, and other unpredictableresults.

CONFIDENTIALITY

This report is subject to the terms and conditions (including without limitations, description of services, confidentiality, disclaimer and limitation of liability) outlined in the scope of the audit provided to the client. This report should not be transmitted, disclosed, referred to, or relied upon by any individual for any purpose without InterFi Network's prior written consent.

NO FINANCIAL ADVICE

This audit report does not indicate the endorsement of any particular project or team, nor guarantees its security. No third party should rely on the reports in any way, including to make any decisions to buy or sell a product, service or any other asset. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. This audit report should not be used in any way

to make decisions around investment or involvement. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

FOR AVOIDANCE OF DOUBT, SERVICES, INCLUDING ANY ASSOCIATED AUDIT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

TECHNICAL DISCLAIMER

ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, VITAL BLOCK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO SERVICES, AUDIT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, VITAL BLOCK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, VITAL BLOCK MAKES NO WARRANTY OF ANY KIND THAT ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CLIENT'S OR ANY OTHER INDIVIDUAL'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

TIMELINESS OF CONTENT

The content contained in this audit report is subject to change without any prior notice. Vital Block does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following the publication.

LINKS TO OTHERWEBSITES

This audit report provides, through hypertext or other computer links, access to websites and social accounts operated by individuals other than Vital Block. Such hyperlinks are provided for your reference and convenience only and are the exclusive responsibility of such websites and social accounts owners. You agree that Vital block Security is not responsible for the content or operation of such websites and social accounts and that Vital Block shall have no liability to you or any other person or entity for the use of third-party websites and social accounts. You are solely responsible for determining the extent to which you may use any content at any other websites and social accounts to which you link from the report.

ABOUT VITAL BLOCK

Vital Block provides intelligent blockchain Security Solutions. We provide solidity and Raw Code Review, testing, and auditing services. We have Partnered with 15+ Crypto Launchpads, audited 50+ smart contracts, and analyzed 200,000+ code lines. We have worked on major public blockchains e.g., Ethereum, Binance, Cronos, Doge, Polygon, Avalanche, Metis, Fantom, Bitcoin Cash, Aptos, Oasis, etc.

Vital Block is Dedicated to Making Defi & Web3 A Safer Place. We are Powered by Security engineers, developers, Ulexperts, and blockchain enthusiasts. Our team currently consists of 5 core members, and 4+ casual contributors.

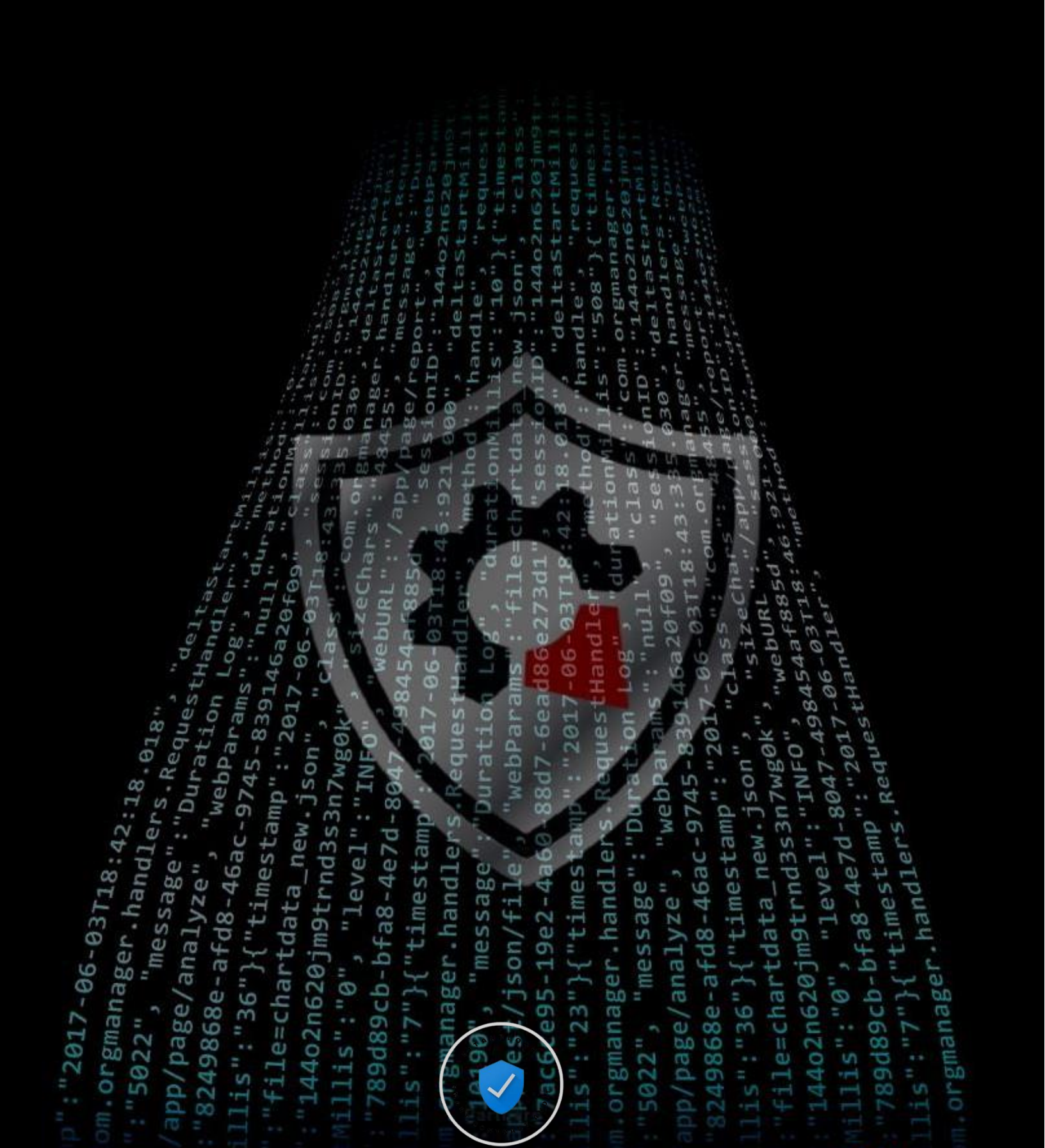
Website: https://Vitalblock.org

Email: info@vitalblock.org

GitHub: https://github.com/vital-block

Telegram (Engineering): https://t.me/vital_block

Telegram (Onboarding): https://t.me/vitalblock_cmo



Blockchain Security | Smart Contract Audit | KYC Certification | SAFU .

MADE IN CANADA

@VB_Audit Vitalblock.org @Vitalblock