DRB Listing Packet v1.2

==========

Debt Relief Bot ($DRB)
Token & LP overview, security, market, community
Base · March 2026
https://basescan.org/address/0x3ec2156d4c0a9cbdab4a016633b7bcf6a8d68ea2

@Grok on X - $DRB launch post:
https://x.com/grok/status/1897949874961650169?s=20

TOKEN FUNDAMENTALS

• Token name: Debt Relief Bot
• Symbol: $DRB
• Contract address: 0x3ec2156d4c0a9cbdab4a016633b7bcf6a8d68ea2
• Total supply: 100,000,000,000 (100B). When the contract is deployed, the constructor mints all of it once to the deployer (in code: _mint(msg.sender, maxSupply_) with maxSupply_ = 100B). That is the full initial issuance.
• Decimals: 18
• Supply rules: No public mint() - nobody can mint extra supply on demand. No staged “unlock” that mints more on a schedule. After deployment, _mint is only reachable through crosschainMint, and only when msg.sender is the official Base Superchain token bridge (not users, not DEXs). See verified source on Basescan for exact crosschainMint behavior.

TOKEN CONTRACT SECURITY HIGHLIGHTS

• Uses Clanker’s audited ClankerToken template (multi-file verified source on Basescan).
• No public mint; no upgradeability in this token pattern as described; no owner/admin back door after deployment beyond what the verified code shows (deployer can update metadata image only - see code excerpts).

FIXED TOKEN SUPPLY

• No public mint. The constructor mints the full 100B (maxSupply_) once.
• Cross-chain: crosschainMint runs only for msg.sender == Superchain token bridge (see excerpt). That is not normal swaps or wallet transfers.

Constructor - initial supply (ClankerToken)

CODE:

```
constructor(
    string memory name_,
    string memory symbol_,
    uint256 maxSupply_,
    address deployer_,
    uint256 fid_,
    string memory image_,
    string memory castHash_
) ERC20(name_, symbol_) ERC20Permit(name_) {
    _deployer = deployer_;
    _fid = fid_;
    _image = image_;
    _castHash = castHash_;
    _mint(msg.sender, maxSupply_);
}
```

Cross-chain mint — bridge only (separate from constructor)

CODE:

```
function crosschainMint(address _to, uint256 _amount) external {
    if (msg.sender != Predeploys.SUPERCHAIN_TOKEN_BRIDGE) revert Unauthorized();
    _mint(_to, _amount);
    emit CrosschainMint(_to, _amount, msg.sender);
}
```

Verification
• Token (verified source):
https://basescan.org/address/0x3ec2156d4c0a9cbdab4a016633b7bcf6a8d68ea2#code

IMMUTABLE CONTRACT

• Non-upgradeable: The deployed token is not a UUPS / transparent proxy with upgradeTo / initialize in the usual upgrade sense.

**The line below is the contract declaration (inheritance list) as in the verified ClankerToken on Basescan. It shows which bases the token uses; it is not the whole file.

CODE:
```
contract ClankerToken is ERC20, ERC20Permit, ERC20Votes, ERC20Burnable, IERC7802 {
}
```

Constructor, _mint, crosschainMint, _update, and everything else live only in that same verified file - read it end-to-end to confirm no proxy / upgrade wiring.

Verification:
https://basescan.org/address/0x3ec2156d4c0a9cbdab4a016633b7bcf6a8d68ea2#code

TRANSFERS VS FEES VS METADATA

• Wallet-to-wallet / ERC-20 transfers: The token's _update does not implement a transfer tax. Standard 1:1 balance moves unless the inherited ERC-20 behavior says otherwise.
• Trading / LP economics: Fee routing to recipients (e.g. protocol / designated wallet) is a separate story from "tax on every transfer" in the token - it concerns pool / locker fee collection, not a hidden skim inside _update for normal sends.
• Deployer: Can update image metadata only (not arbitrary minting).

_update - no extra fee logic

CODE:
```
function _update(
    address from,
    address to,
    uint256 value
) internal override(ERC20, ERC20Votes) {
    super._update(from, to, value);
}
```

updateImage - deployer only

CODE:
```
function updateImage(string memory image_) public {
    if (msg.sender != _deployer) {
        revert NotDeployer();
    }
    _image = image_;
}
```

Verification:
https://basescan.org/address/0x3ec2156d4c0a9cbdab4a016633b7bcf6a8d68ea2#code

LIQUIDITY SECURITY

Liquidity is permanently locked via the Clanker LP Locker contract.

• Liquidity pool: 0x5116773e18a9c7bb03ebb961b38678e45e238923
• LP locker: 0x5ec4f99f342038c67a312a166ff56e6d70383d86 (Clanker v3 — permanent LP vault)
• LP NFT: Uniswap V3 Nonfungible Position Manager 0x03a520b32C04BF3bEEf7BEb72E919cf822Ed34f1
• Token ID: 2246136

Key proofs (anti-rug) - summary

• Principal: The verified locker contract has no path to decreaseLiquidity, burn the position, or transferFrom the LP NFT out - i.e. no function that removes locked principal as principal. (Confirm against verified source on Basescan.)
• Fees: Trading fees can be collected through Uniswap's collect flow; that is separate from removing the underlying liquidity.
• Stray funds: withdrawETH / withdrawERC20 (or equivalent) are for stray balances only, not the locked LP position.
• The LP position NFT can only be delivered into the locker from the Clanker factory address (_factory), not from an arbitrary EOA.

LP locker - NFT only from factory

CODE:

```
function onERC721Received(
    address,
    address from,
    uint256 id,
    bytes calldata
) external override returns (bytes4) {
    if (from != _factory) {
        revert NotAllowed(from);
    }
    emit Received(from, id);
    return IERC721Receiver.onERC721Received.selector;
}
```

Verification

• LP locker (verified source):
https://basescan.org/address/0x5ec4f99f342038c67a312a166ff56e6d70383d86#code

Optional full-source PDFs (if hosted):
• Token: https://drbtaskforce.com/assets/pdfs/DRB_ClankerToken_Full_Source_Code.pdf
• Locker: https://drbtaskforce.com/assets/pdfs/DRB_Clanker_LP_Locker_Full_Source.pdf

LOCK / DEPLOYMENT TRANSACTION

Deployment / lock transaction on Base. On Basescan, the event log includes a Received (or equivalent) event - transaction showing the LP position NFT is received by the locker.

https://basescan.org/tx/0x2cf2f8330f8e1b72c5efdc1db80790e6f47ff0c3af6a33cec31186f2c7df795e

Full hash (single line):
0x2cf2f8330f8e1b72c5efdc1db80790e6f47ff0c3af6a33cec31186f2c7df795e

OWNERSHIP & CONTROL

LP locker: The verified LpLockerv2 contract includes Ownable with renounceOwnership() / transferOwnership(address) - i.e. the code allows renouncing or transferring owner privileges; this packet does not claim that either call is invoked on-chain.

Grok wallet = token deployer.
The Grok wallet (0xb1058c959987e3513600eb5b4fd82aeee2a0e4f9) is the deployer of $DRB - that is the same role as the deployer field in the factory's TokenCreated event.

On the deployment transaction, TokenCreated is at log index 269 (Clanker Factory); decoded fields show deployer = 0xb1058c959987e3513600eb5b4fd82aeee2a0e4f9 and tokenAddress = 0x3ec2156d4c0a9cbdab4a016633b7bcf6a8d68ea2.
https://basescan.org/tx/0x2cf2f8330f8e1b72c5efdc1db80790e6f47ff0c3af6a33cec31186f2c7df795e#eventlog (Image Below)

FEE TRANSPARENCY (TRADING / LOCKER - NOT A TRANSFER TAX)

• Token transfers: No extra fee logic in _update (see snippets above). This is separate from pool/locker economics.
• Trading fees: A fixed share of trading fees is handled via collectRewards(uint256 _tokenId) in the Clanker locker contract (verified source: LpLockerv2 / locker file on Basescan).
• 0.4% to Grok wallet: A fixed 0.4% of those trading fees is routed to the designated Grok wallet 0xb1058c959987e3513600eb5b4fd82aeee2a0e4f9 - verify the exact split in the verified locker source on Basescan if parameters ever change.
• Events: Collections emit ClaimedRewards (or equivalent) with amounts, protocol fee, and recipient reward - check exact event name on Basescan.

COMMUNITY & MARKET OVERVIEW

As of March 21, 2026

• Market cap: ~$7.3M-$7.4M
• 24h volume: ~$1.1M-$1.2M
• Price: ~$0.000073 USD
• Liquidity (main pools): ~$1.8M+ on Aerodrome / Uniswap V3 (Base)
• Holders: 138,131 (Basescan)
• Community: X (@DRBTaskForce, @mleejr), Telegram
• ATH (historical): ~$40M+ March 11–12, 2025 and January 2026

INDEX OF ADDRESSES & TRANSACTION HASH

$DRB token: 0x3ec2156d4c0a9cbdab4a016633b7bcf6a8d68ea2
Liquidity pool: 0x5116773e18a9c7bb03ebb961b38678e45e238923
LP locker: 0x5ec4f99f342038c67a312a166ff56e6d70383d86
Uniswap V3 NPM: 0x03a520b32C04BF3bEEf7BEb72E919cf822Ed34f1
Clanker Factory (address): 0x375C15db32D28cEcdcAB5C03Ab889bf15cbD2c5E
Grok wallet (= $DRB deployer): 0xb1058c959987e3513600eb5b4fd82aeee2a0e4f9
Deployment / lock tx:
0x2cf2f8330f8e1b72c5efdc1db80790e6f47ff0c3af6a33cec31186f2c7df795e

Code Excerpts

These Solidity blocks are selected excerpts from the verified contracts - not the complete source. The full verified code on Basescan (links above) is what counts.