

# The verge is (not) going to break your contracts!

Hadrien Croubois (Amxx)

OpenZeppelin





# OpenZeppelin's thesis

- There will be a **trillion dollar open economy** built on blockchains and **powered by smart contracts**
- This new, open economy will be **built by teams of creative people developing new applications used by billions of people**
- These teams **will need a set of tools, products and services** to make sure that what they are building is **safe and reliable**
- OpenZeppelin will be a **leading provider of these solutions**, allowing teams to **build faster with lower risk**

# Contracts

@openzeppelin/contracts@5.1.0

@openzeppelin/contracts-upgradeable@5.1.0



A library of modular, reusable, secure smart contracts for the Ethereum network, written in Solidity.

- ✓ Leverage standard, tested, and community-reviewed contracts.
- ✓ Most popular library in the industry.
- ✓ Learn from best practices adopted by the ecosystem.
- ✓ Reduce your attack surface by reusing audited code.

VISIT SITE

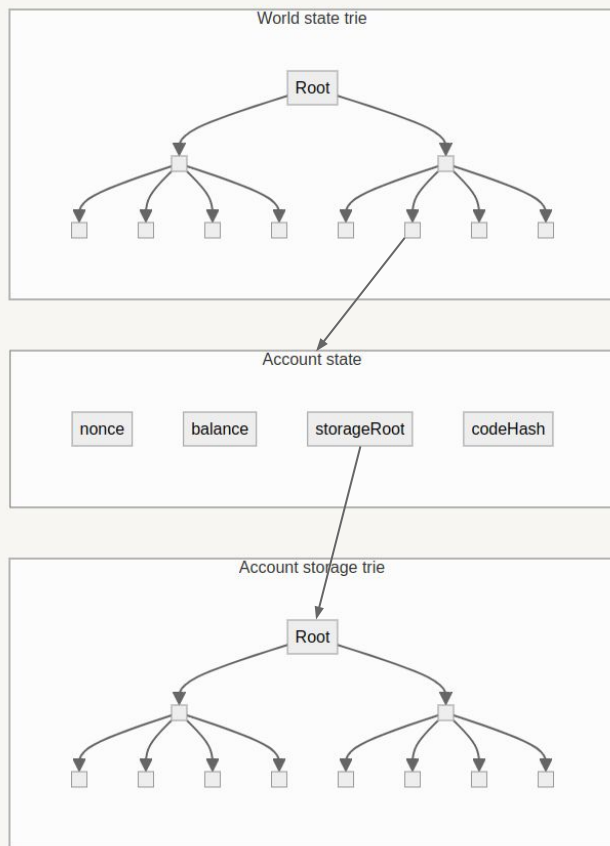
GO TO DOCS



## Section 1

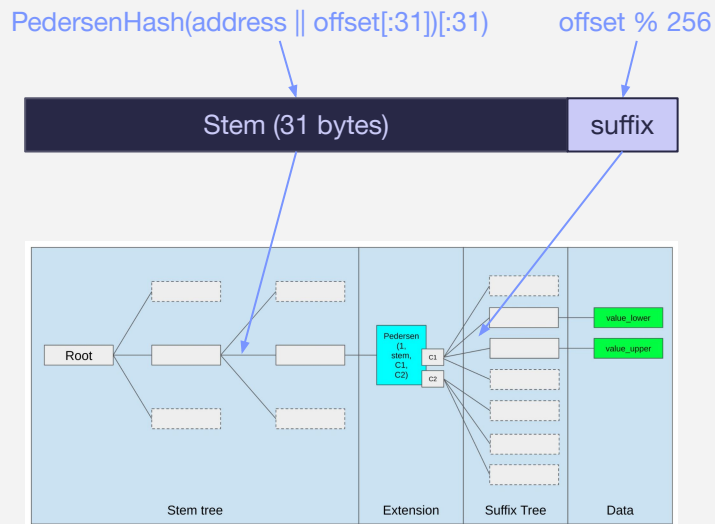
# How is the verge going to affect devs?





## How is smart contract state stored?

- Each account comes with a storage trie
- **Used for persistent data**  
ERC-20 balances, ERC-721 ownership, everything
- **Storage is divided in slots**  
Each account has  $2^{256}$  slots available
- **Slots must be “warmed up” before use**  
You only pay the warm-up price once per tx
- All slots are identical



## That all changes with the verge

- All storage in the same (verkle) trie
- Position in that shared trie is a combination of account and offset ("slot number")
- Slots are gathered in buckets (aka extensions)  
For a given account, consecutive slots are in the same bucket (up to 256 slots)
- Buckets are "warmed up", not slots  
So reusing slots in a given bucket is efficient



# Here's the timeline.

**Dencun**



March 13th 2024

**Pectra**



~ April 2025

**The Verge**



~2026?



## Section 2

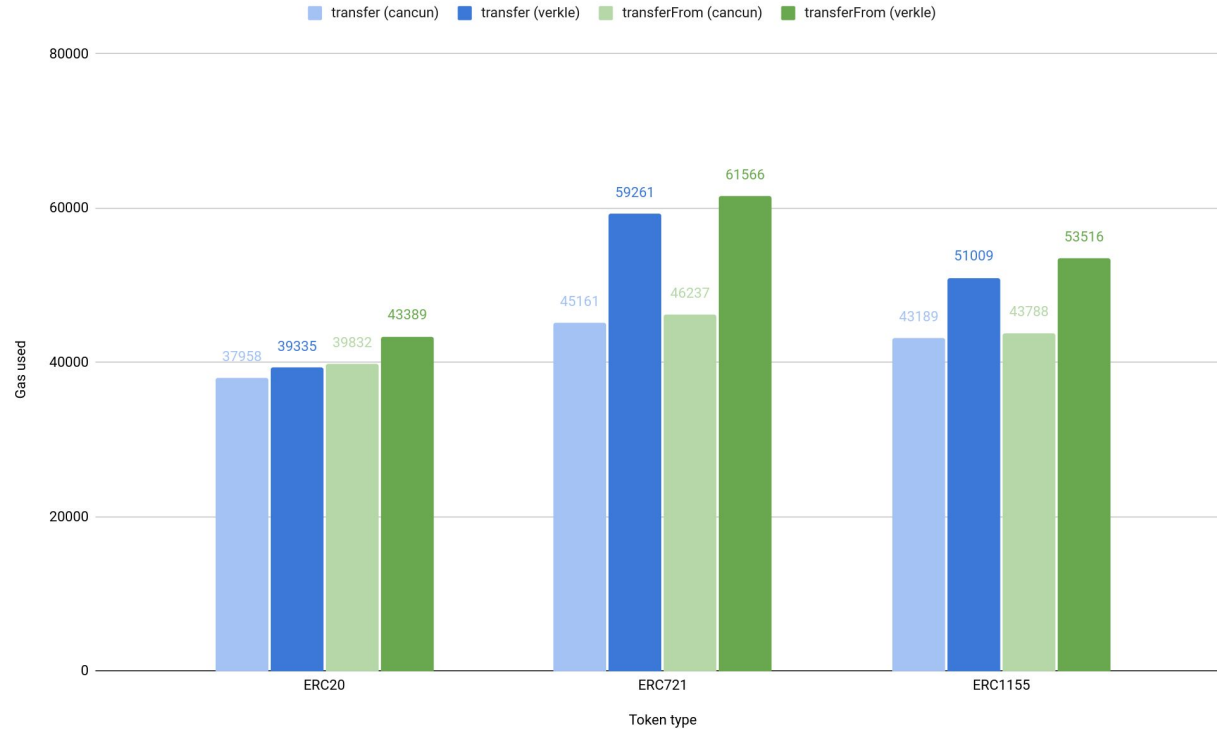
# **How will it impact existing contracts?**



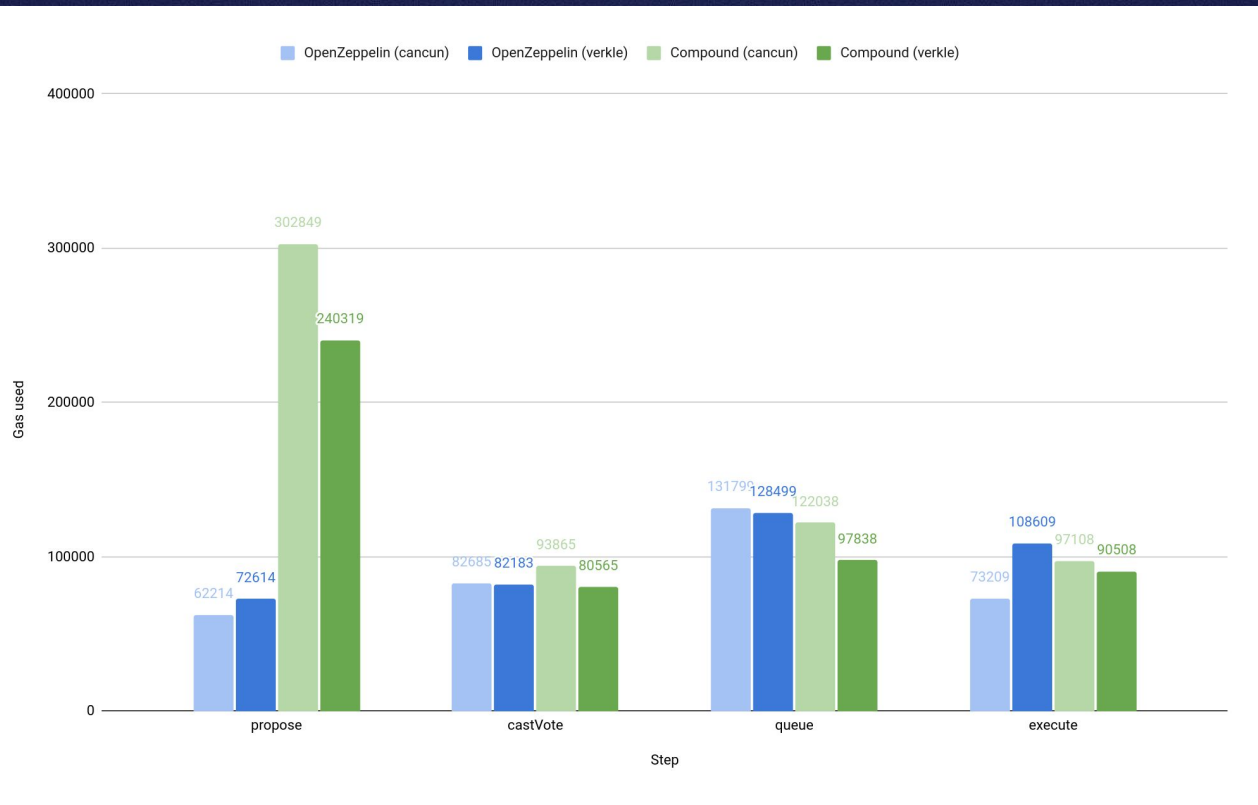
# Disclaimer

Gas numbers shown in this presentation are NOT final.  
Verkle implementation is still in progress.

# Common tokens

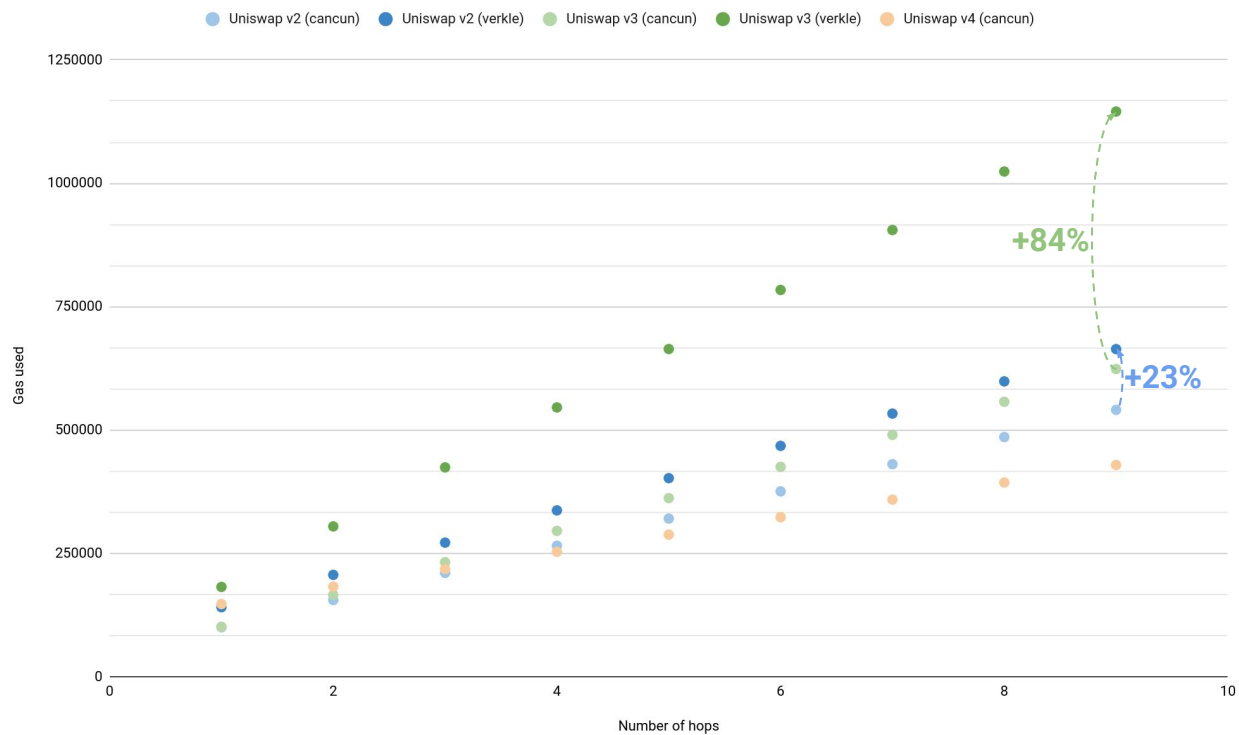


# Governance (with Timelock)





# Token swaps

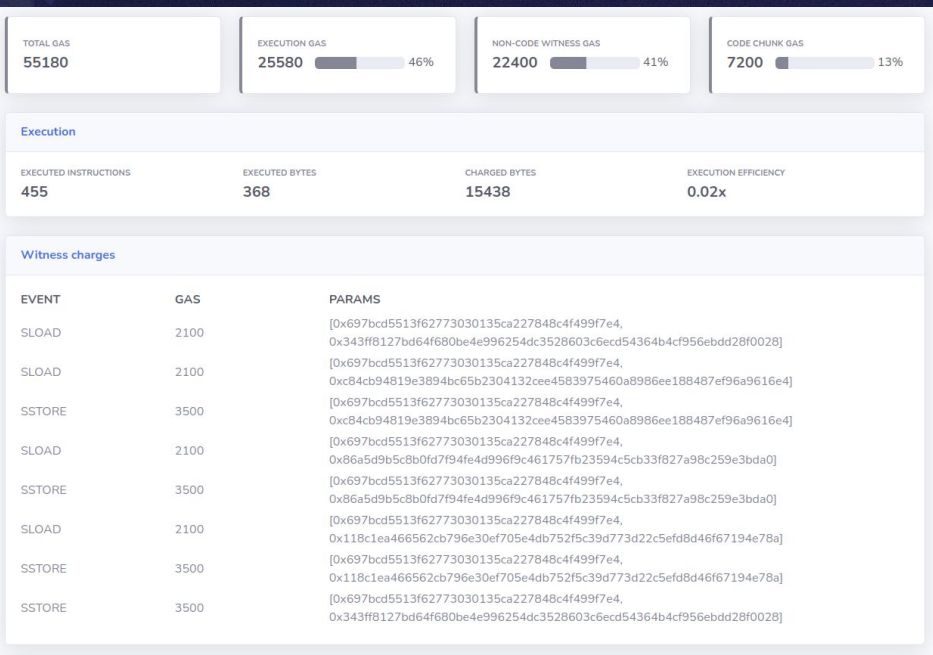


Section 3

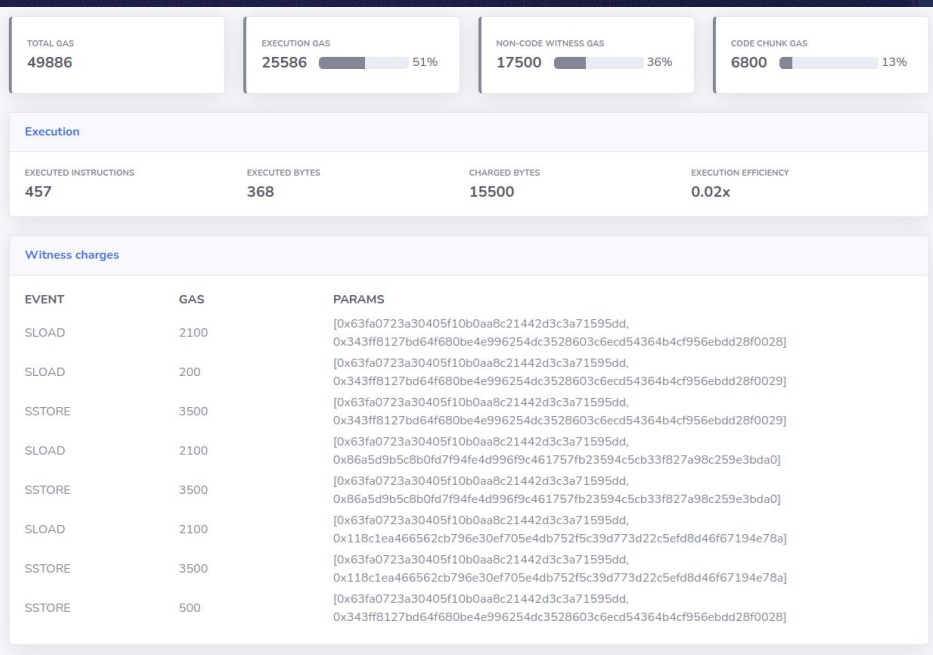
# Smart contract design for a Verkle-EVM

```
contracts/token/ERC721/ERC721.sol
25 25 // Token symbol
26 26 string private _symbol;
27 27
28 - mapping(uint256 tokenId => address) private _owners;
28 + struct TokenDetails {
29 +     address owner;
30 +     address approval;
31 + }
29 32
30 - mapping(address owner => uint256) private _balances;
33 + struct AccountDetails {
34 +     uint256 balance;
35 +     mapping(address => bool) operators;
36 + }
31 37
32 - mapping(uint256 tokenId => address) private _tokenApprovals;
38 + mapping(uint256 tokenId => TokenDetails) private _tokens;
33 39
34 - mapping(address owner => mapping(address operator => bool)) private _operatorApprovals;
40 + mapping(address owner => AccountDetails) private _accounts;
35 41
36 42 /**
37 43  * @dev Initializes the contract by setting a 'name' and a 'symbol' to the token collection.
44 44
45 45
46 46
47 47
48 48
49 49
50 49
51 49
52 49
53 49
54 49
55 49
56 49
57 49
58 64 if (owner == address(0)) {
59 65     revert ERC721InvalidOwner(address(0));
60 66 }
61 - return _balances[owner];
67 + return _accounts[owner].balance;
62 68 }
63 69
64 70 /**
71 71
72 71
73 71
74 71
75 71
76 71
77 71
78 71
79 71
80 71
81 71
82 71
83 71
84 71
85 71
86 71
87 71
88 71
89 71
90 71
91 71
92 71
93 71
94 71
95 71
96 71
97 71
98 71
99 71
100 71
101 71
102 71
103 71
104 71
105 71
106 71
107 71
108 71
109 71
110 71
111 71
112 71
113 71
114 71
115 71
116 71
117 71
118 71
119 71
120 71
121 71
122 71
123 71
124 71
125 71
126 71
127 71
128 134 * @dev See {IERC721-IsApprovedForAll}.
129 135 */
130 136 function isApprovedForAll(address owner, address operator) public view virtual returns (bool) {
131 - return _operatorApprovals[owner][operator];
137 + return _accounts[owner].operators[operator];
132 138 }
```



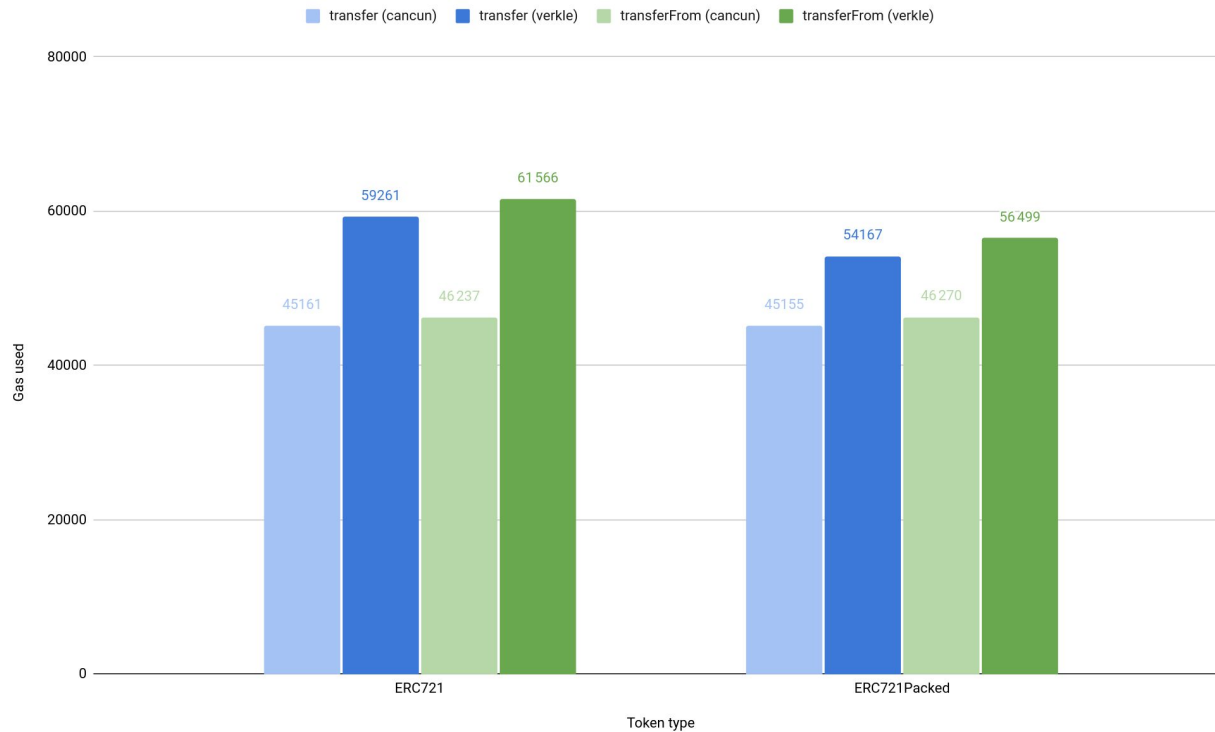


transferFrom on a "Normal ERC-721"



transferFrom on a "Packed ERC-721"

# ERC-721 Packed



Section 4

# Conclusion



## The verge:

- is NOT going to break you contracts
- will impact the gas costs of all contracts, including the “old” ones.
- should be taken into consideration when designing/optimizing you next contracts!

## You can act now:

- avoid “persistent” storage (transient can help)
- use contiguous storage
  - favor arrays to mapping
  - consolidate mappings using structs
  - store the length of the arrays at the first position ([see my EthCC talk](#))
- (if possible) align your structs with extensions
  - see ERC-7201
- change the language/compiler.

# Thank you!

**Hadrien Croubois**

OpenZeppelin

hadrien@openzeppelin.com

@Amxx

<https://shared.hadriencroubois.com/slides/Devcon2024-verge.pdf>