



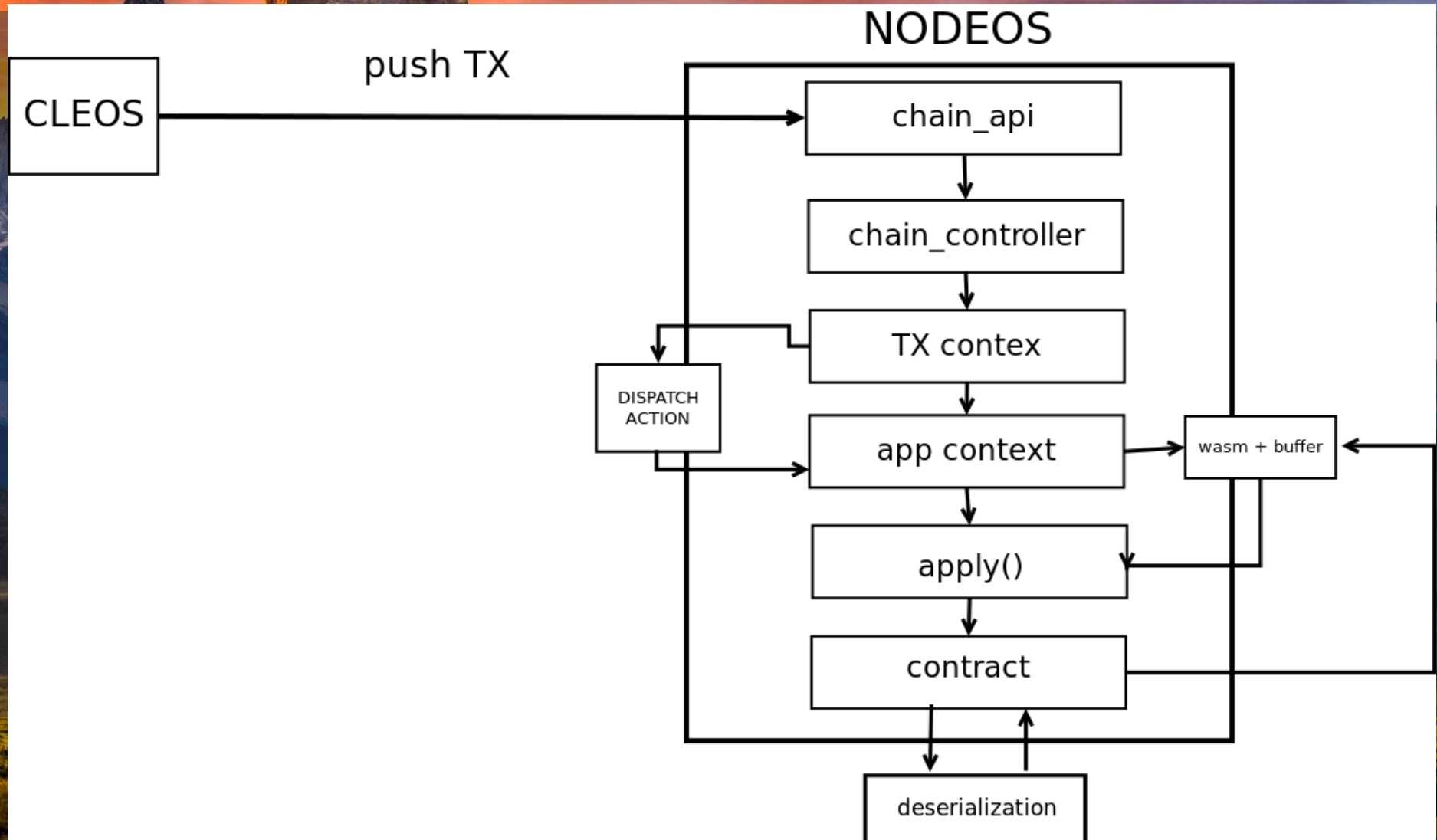
EOS ARGENTINA

EOS SMART CONTRACTS

KEY CONCEPTS:

- * WEBASSEMBLY BASED**
- * CPU, RAM + BANDWIDTH**
- * LIBRARY C++ EOSIOLIB**
- * NODEOS Y CLEOS**

EOS SMART CONTRACTS



INTRINSICS

ORGINIZED IN GROUPS ACCORDING TO THEIR USE:

- * MATH_API
- * DATABASE_API
- * CRYPTO_API
- * STRING_API
- * SYSTEM_API
- * ACTION_API
- * TRANSACTION_API
- ETC

INTRINSICS: DATABASE

THIS INTRINSICT LETS YOU INTERACT WITH THE INTERNAL STORAGE, WHERE CONTRACTS SAVE THE STATE.

INTRINSICS: DATABASE

THIS INTRINSICT LETS YOU INTERACT WITH THE INTERNAL STORAGE, WHERE CONTRACTS SAVE THE STATE.

INTRINSICS: DATABASE

THEY ARE GROUP BY:

- * READ, WRITE, UPDATE_

- * INDEX MANAGEMENT_

EOSIOLIB

**EOSIOLIB IS A LIBRARY THAT LET YOU INTERACT
WITH INTRINSICTS IN A EASY WAY.**

IT ENCAPSULATES THE COMPLEXITY OF INTERNAL APIS

EOSIOLIB (DICE)

```
//@abi table game i64
struct game {
    uint64_t id;
    asset    bet;
    eosio::time_point_sec deadline;
    player   player1;
    player   player2;

    uint64_t primary_key()const { return id; }

    EOSLIB_SERIALIZE( game, (id)(bet)(deadline)(player1)(player2) )
};

typedef eosio::multi_index< N(game), game> game_index;

//@abi table global i64
struct global_dice {
    uint64_t id = 0;
    uint64_t nextgameid = 0;

    uint64_t primary_key()const { return id; }

    EOSLIB_SERIALIZE( global_dice, (id)(nextgameid) )
};

typedef eosio::multi_index< N(global), global_dice> global_dice_index;
```


EOSIO LIB

* MULTI INDEX

* CRYPTO

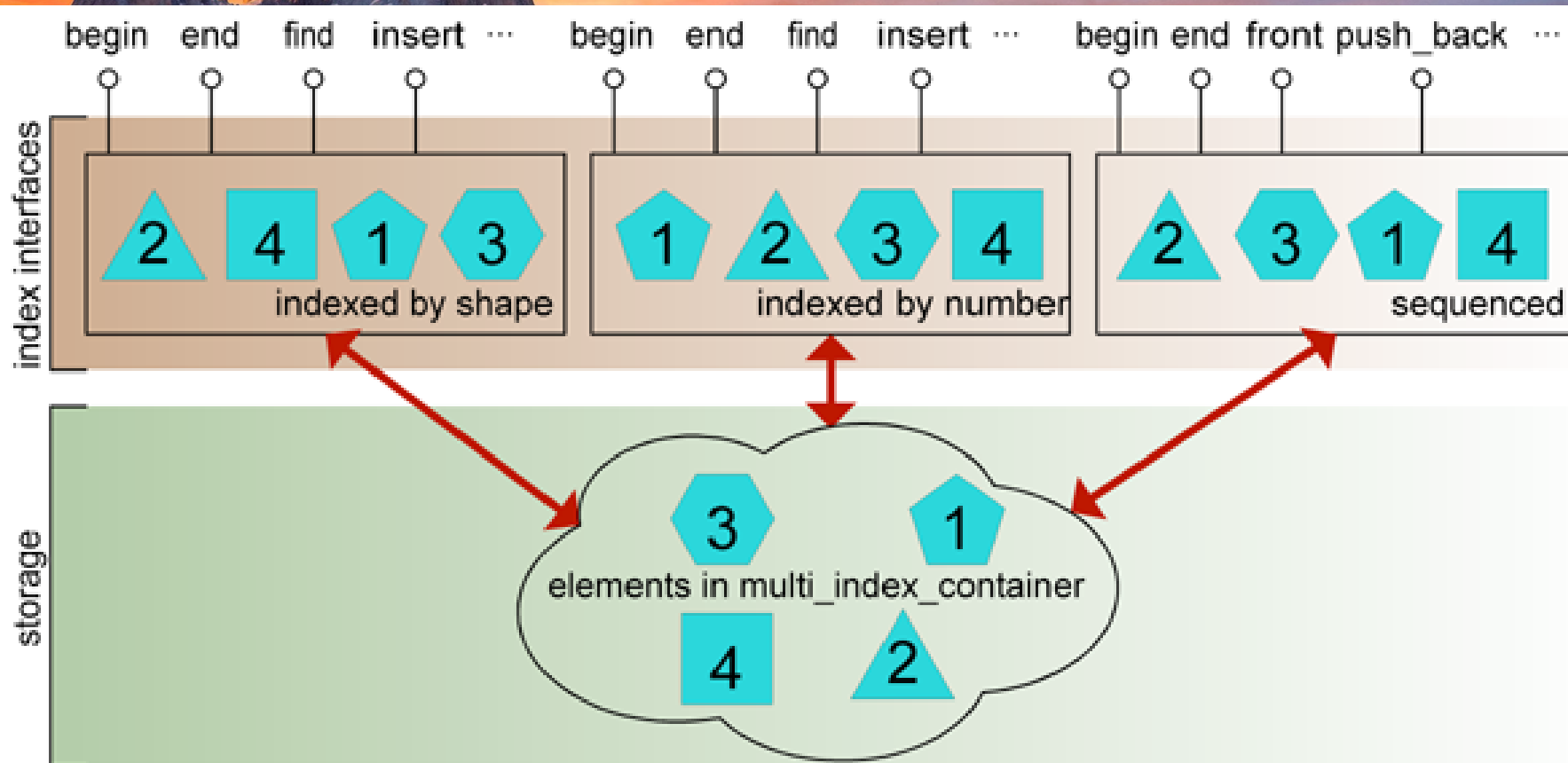
* DISPATCHER

* SERIALIZE

MULTIINDEX

**LETS YOU WORK WITH A CONTAINER OF OBJECTS
BUT ARRANGED WITH MULTIPLES INDEX.**

MULTIINDEX



DISPATCHER (EOSIOLIB)

- * HELPS YOU TO DESERIALIZE THE BINARY
- * HELPS THE ABI GENERATOR
- * DECLARES THE CONTRACT CLASS AND ACTIONS
- * ARRANGE THE PARAMETERS AND CALLS THE FUNCTION

SERIALIZATION (SERIALIZE)

```
→ gameoflife-ts git:(master) x cleos push action -j -d -s gameoflifets create \
    '["gameoflifets","game3",10,10,20]' \
    -p gameoflifets
{
  "expiration": "2018-06-09T23:36:07",
  "ref_block_num": 540,
  "ref_block_prefix": 2985976104,
  "max_net_usage_words": 0,
  "max_cpu_usage_ms": 0,
  "delay_sec": 0,
  "context_free_actions": [],
  "actions": [{
    "account": "gameoflifets",
    "name": "create",
    "authorization": [{
      "actor": "gameoflifets",
      "permission": "active"
    }]
  }, {
    "data": "80b35a2e2eaaa4610000000080a1a4610a0000000a00000014000000"
  }],
  "transaction_extensions": [],
  "signatures": [],
  "context_free_data": []
}
```


SERIALIZATION (SERIALIZE)

80B35A2E2EAAA461 0000000080A1A461 0A000000
0A000000 14000000

GOES TO:

GAMEOFLIFETS GAME3 10 10 20

DESERIALIZATION(DISPATCHER)

GAMEOFLIFETS GAME3 10 10 20

GOES TO:

80B35A2E2EAAA461 0000000080A1A461 0A000000
0A000000 14000000

TYPESCRIPT

**“IT IS A STRICT SYNTACTICAL SUPERSSET OF JAVASCRIPT,
AND ADDS OPTIONAL STATIC TYPING TO THE LANGUAGE.”
WIKIPEDIA**

ADDS VALUE TO THE SMART CONTRACTS ECOSYSTEM

ASSEMBLYSCRIPT

**“ A TYPESCRIPT TO WEBASSEMBLY COMPILER
[HTTP://ASSEMBLYSCRIPT.ORG](http://assemblyscript.org) “**

MADE BY DCODEIO

ASSEMBLYSCRIPT

```
import "allocator/arena";
import { printstr } from "../eoslib/utils";

export function apply(receiver: u64, code: u64, action: u64): void {
    printstr("Hello EOS from TS");
}
```

```
by eosio [txs: 0, lib: 16, confirmed: 0]
3078501ms thread-0    producer_plugin.cpp:1073    produce_block    ] Produced b
by eosio [txs: 0, lib: 17, confirmed: 0]
3078852ms thread-0    apply_context.cpp:28    print_debug    ]
[(gameoflifets,removeall)->gameoflifets]: CONSOLE OUTPUT BEGIN =====
Hello EOS from TS
[(gameoflifets,removeall)->gameoflifets]: CONSOLE OUTPUT END =====
3078861ms thread-0    apply_context.cpp:28    print_debug    ]
[(gameoflifets,create)->gameoflifets]: CONSOLE OUTPUT BEGIN =====
```


GAME OF LIFE C++

IT WAS A POC MADE BY TODD FLEMING

[HTTPS://GITHUB.COM/TBFLEMING/EOS-GAME-OF-LIFE](https://github.com/tbfleming/eos-game-of-life)

GAME OF LIFE C++

```
game_of_life.cpp x
64
65 struct gameoflife : eosio::contract {
66     gameoflife(account_name self) : contract{self} {}
67
68     void removeall(account_name user) {
69         require_auth(user);
70
71         // multi_index can't erase when the format changed
72         auto it = db_lowerbound_i64(_self, user, N(boards), 0);
73         while (it >= 0) {
74             auto del = it;
75             uint64_t dummy;
76             it = db_next_i64(it, &dummy);
77             db_remove_i64(del);
78         }
79     }
80
81     void remove(account_name user, eosio::name game) {
82         require_auth(user);
83         auto it = db_find_i64(_self, user, N(boards), game);
84         if (it >= 0)
85             db_remove_i64(it);
86     }
87
88     void create(account_name user, eosio::name game, uint32_t num
89                 uint32_t num_cols, uint32_t seed) {
90         remove(user, game);
```


GAME OF LIFE C++

```
65 ],
66 "actions": [{
67     "name": "removeall",
68     "type": "removeall",
69     "ricardian_contract": ""
70 }, {
71     "name": "remove",
72     "type": "remove",
73     "ricardian_contract": ""
74 }, {
75     "name": "create",
76     "type": "create",
77     "ricardian_contract": ""
78 }, {
79     "name": "step",
80     "type": "step",
81     "ricardian_contract": ""
82 }
83 ],
84 "tables": [{
85     "name": "boards",
86     "index_type": "i64",
87     "key_names": [
88         "game"
89     ],
90     "key_types": [
91         "name"
92     ],
93     "type": "board"
94 }
95 ],
```


GAME OF LIFE TYPESCRIPT

```
13 /***** GAMEOFLIFE CLASS *****/
14 export class GameOfLife extends Contract {
15
16     dummy: u64;
17
18     // step action
19     on_step(args: Step): void {
20         EOS.dumptime(1,0);
21         EOS.require_auth(args.user);
22
23         let it = EOS.db_find_i64(this.receiver, args.user, N("boards"), a
24         assert(it >= 0, "game not found");
25
26         let len = EOS.db_get_i64(it, 0, 0);
27         assert(len >= 0, "invalid length");
28
29         let arr = new Uint8Array(len);
30         len = EOS.db_get_i64(it, <usize>arr.buffer, len);
31         assert(len >= 0, "invalid length");
32
33         let ds = new DataStream(<usize>arr.buffer, len);
34         let old = Board.from_ds(ds);
35
36         EOS.dumptime(2,0);
37
38         let bsize = old.get_size();
39
40         let board = new Board();
41         board.game = old.game;
```


GAME OF LIFE ASSEMBLYSCRIPT

TO MAKE ASSEMBLYSCRIPT COMPATIBLE WITH EOS SMART CONTRACTS WE MODIFY IT , SO IT DOESN'T CALL UNSUPPORTED ABORT INTRINSIC.

```
2755 const abortInternalName = "abort";
2756
2757 /** Compiles an abort wired to the conditionally imp
2758 export function compileAbort(
2759   compiler: Compiler,
2760   message: Expression | null,
2761   reportNode: Node
2762 ): ExpressionRef {
2763   return compiler.module.createUnreachable();
2764
2765   // var program = compiler.program;
2766   // var module = compiler.module;
2767
2768   // var stringType = program.typesLookup.get("strin
2769   // if (!stringType) return module.createUnreachabl
2770
2771   // var abortPrototype = program.elementsLookup.get
2772   // if (!abortPrototype || abortPrototype.kind != E
2773
```


PERFORMANCE

**SETUP: WE CREATE 2 ACCOUNTS IN A LOCAL TESTNET
1 FOR TYPESCRIPT VERSION, 1 FOR C++**

PERFORMANCE

```
cleos system newaccount \
  --buy-ram-kbytes 200 \
  --stake-net "100.0000 EOS" \
  --stake-cpu "100.0000 EOS" \
  eosio \
  gameoflifets \
  EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV \
  EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV

cleos set contract gameoflifets ~/dev/gameoflife-ts/gameoflife-ts -p gameoflifets

cleos system newaccount \
  --buy-ram-kbytes 200 \
  --stake-net "100.0000 EOS" \
  --stake-cpu "100.0000 EOS" \
  eosio \
  gameoflifecc \
  EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV \
  EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV

cleos set contract gameoflifecc ~/dev/game_of_life -p gameoflifecc
```


PERFORMANCE (c++)

```
→ gameoflife-ts git:(master) x cleos get account gameoflifecc
permissions:
  owner      1:      1 EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
  active     1:      1 EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
memory:
  quota:      199 Kb      used:      140 Kb
net bandwidth:
  delegated:   100.0000 EOS      (total staked delegated to account from others)
  used:        6.626 Kb
  available:   88.97 Gb
  limit:       88.97 Gb
cpu bandwidth:
  delegated:   100.0000 EOS      (total staked delegated to account from others)
  used:        958 us
  available:   5.061 hr
  limit:       5.061 hr
```


PERFORMANCE (Ts)

```
→ gameoflife-ts git:(master) x cleos get account gameoflifets
permissions:
  owner      1:      1 EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
  active     1:      1 EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
memory:
  quota:      199 Kb      used:      40.39 Kb
net bandwidth:
  delegated:   100.0000 EOS      (total staked delegated to account from others)
  used:        1.423 Kb
  available:   118.7 Gb
  limit:       118.7 Gb
cpu bandwidth:
  delegated:   100.0000 EOS      (total staked delegated to account from others)
  used:        558 us
  available:   6.747 hr
  limit:       6.747 hr
```


PERFORMANCE (c++)

```
{
  "rows": [{
    "game": "game3",
    "rows": [
      "  **  ** ",
      "** * * **",
      " **  ** ",
      " **   ***",
      " * *** ** ",
      "*** *****",
      "   *****",
      " ** * ***",
      "   **  **",
      "  * *  * "
    ]
  }
],
"more": false
}
```

```
executed transaction: 656c780cc8c2c350e0acc793c921abf5a76ded1c725b0bc2e37f3dcc8bbeb7bc3 112 bytes 887 us
warning: transaction executed locally, but may not be confirmed by the network yet
```

```
^C
```


PERFORMANCE (Ts)

```
{
  "rows": [{
    "game": "game3",
    "rows": [
      " *      ",
      " *      * ",
      " *      * **",
      " ",
      " ",
      " *      * ",
      "*** * ",
      "*** *  * ",
      "** *   ",
      "** **  * "
    ]
  }
],
  "more": false
}
```

```
executed transaction: bca8be13144642a2e767f83382c92e1368d9b562a00861758913dbac138f8de1
warning: transaction executed locally, but may not be confirmed by the network yet
```

112 bytes 10619 us

PERFORMANCE (NODEOS)

```
474  
475     fc::time_point start = fc::time_point::now();  
476  
477     void dumptime(uint64_t step, uint64_t sub_step) {  
478         auto elapsed = fc::time_point::now() - start;  
479         ilog("STEP${s}.${ss}: ${t}", ("s", step)("ss", sub_step)("t", elapsed));  
480     }  
481  
482
```


PERFORMANCE (C++)

```
void step(account_name user, eosio::name game) {  
    dumptime(1,0);  
    require_auth(user);  
    eosio::multi_index<N(boards), board> boards(_self, user);  
    const board& b = boards.get(game);  
    dumptime(2,0);  
    boards.modify(b, user, ::step);  
    dumptime(3,0);  
};
```

```
EOSIO_ABI(gameoflife, (removeall)(remove)(create)(step))
```

```
pp:1073    produce_block    ] Produced block 00000019b1  
0]  
pp:1073    produce_block    ] Produced block 0000001a5c  
0]  
:479       dumptime        ] STEP1.0: 44  
:479       dumptime        ] STEP2.0: 352  
:479       dumptime        ] STEP3.0: 731  
pp:1073    produce_block    ] Produced block 0000001b6c  
0]  
pp:1073    produce_block    ] Produced block 0000001c6d  
0]
```


PERFORMANCE (Ts)

```
on_step(args: Step): void {
    EOS.dumptime(1,0);
    EOS.require_auth(args.user);

    let it = EOS.db_find_i64(this.receiver, args.user);
    assert(it >= 0, "game not found");

    let len = EOS.db_get_i64(it, 0, 0);
    assert(len >= 0, "invalid length");

    let arr = new Uint8Array(len);
    len = EOS.db_get_i64(it, <usize>arr.buffer, len);
    assert(len >= 0, "invalid length");

    let ds = new DataStream(<usize>arr.buffer, len);
    let old = Board.from_ds(ds);

    EOS.dumptime(2,0);
```

```
73      produce_block      ] Produced block 0000
                                ] STEP1.0: 114
                                ] STEP2.0: 383
                                ] STEP3.0: 10243
73      produce_block      ] Produced block 0000
73      produce_block      ] Produced block 0000
```


OPTIMIZATION (PoC TS)

```
export class Board implements ISerializable {  
  game: u64;  
  rows: string[];  
  
  static from_ds(ds: DataStream): Board {  
    let game: u64 = ds.read<u64>();
```

BEFORE

```
1 export class Board implements ISerializable {  
2   game: u64;  
3   rows: Array<Array<u8>>;  
4  
5   static ASTERIX : u8 = 42;  
6   static SPACE   : u8 = 32;  
7
```

AFTER

```
0]  
pp:1073      produce_block      ] Produced block 000000  
0]  
:479         dumptime           | STEP1.0: 114  
:479         dumptime           | STEP2.0: 369  
:479         dumptime           | STEP3.0: 4342  
pp:1073      produce_block      | Produced block 000000  
0]  
pp:1073      produce_block      ] Produced block 000000  
0]
```


CONCLUSIONS

*** C++ IS HARD TO MATCH
BUT FURTHER OPTIMIZATION CAN GIVE
SATISFACTORY RESULTS**

**THERE IS ALREADY AN ONGOIN EFFORT
TO HAVE AN EOSIOLIB ON TYPESCRIPT**

[HTTPS://GITHUB.COM/TOONSEVRIN](https://github.com/toonsevrin)

[HTTPS://GITHUB.COM/EOS-TYPESCRIPT/EOS-ASSEMBLYSCRIPT-API](https://github.com/eos-typescript/eos-assemblyscript-api)



谢谢