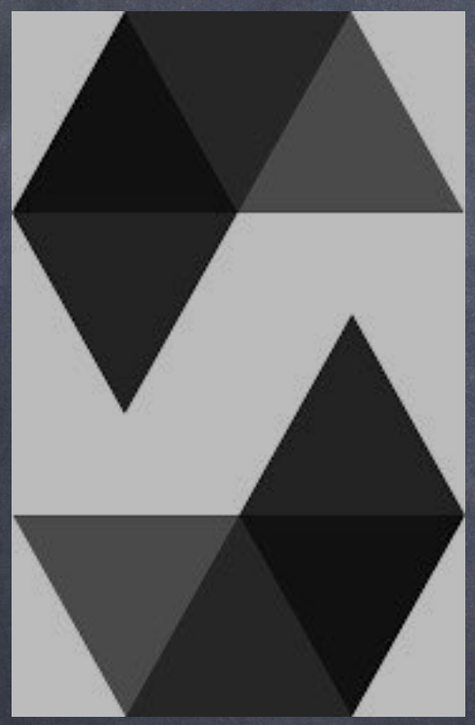
Solidity 201

Secureum Bootcamp

#102

Inheritance



Multiple Inheritance
Polymorphism

Base -> Derived
Single Contract

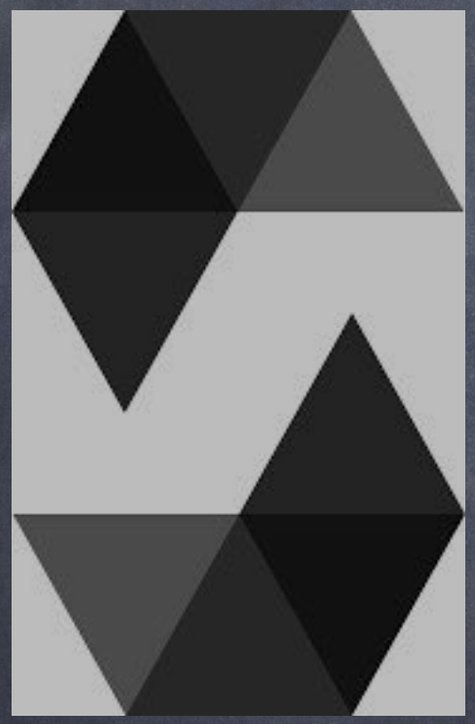
Function Overriding
virtual -> override

Diamond Problem
C3 Linearization

#103

Contract Types

Contract/Abstract/
Interface/Library



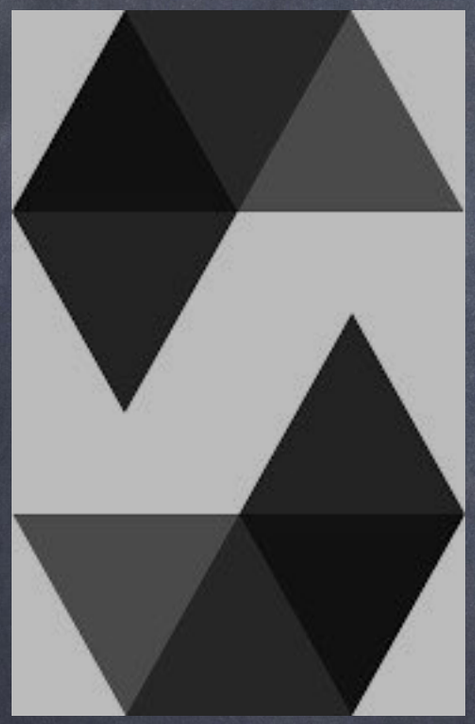
abstract: Unimplemented
Function -> At Least One

interface: None, external
No Constructor/State Vars

Library: Deployed Once,
delegatecall

#104

Using for



Directive
using A for B

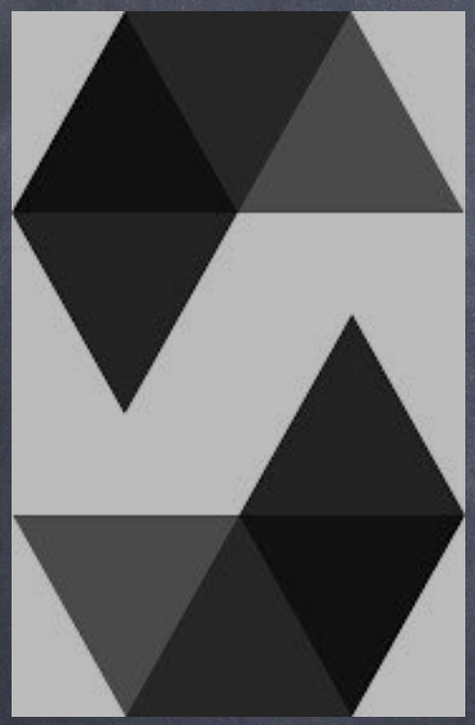
Attach Functions
Library A -> Type B

Object -> First Parameter
Only Inside Contract

E.g.: using SafeMath for
uint256;

#105

Base Class Functions



Inheritance Hierarchy

Derived -> Base

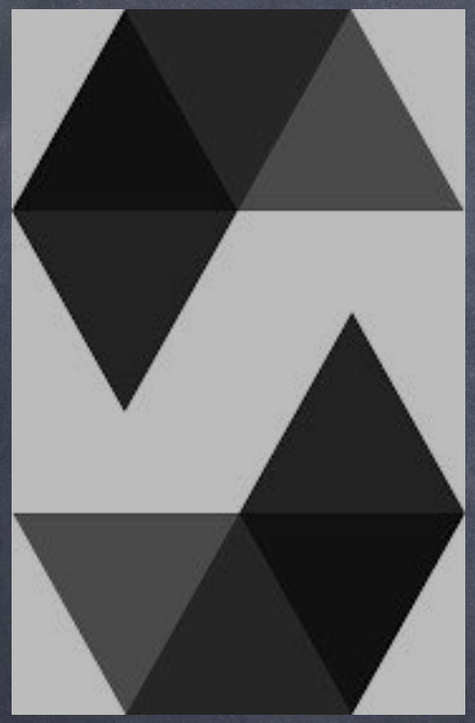
Explicit
Contract.function()

One Level Up
super.function()

#106

Shadowing

State Variable Shadowing



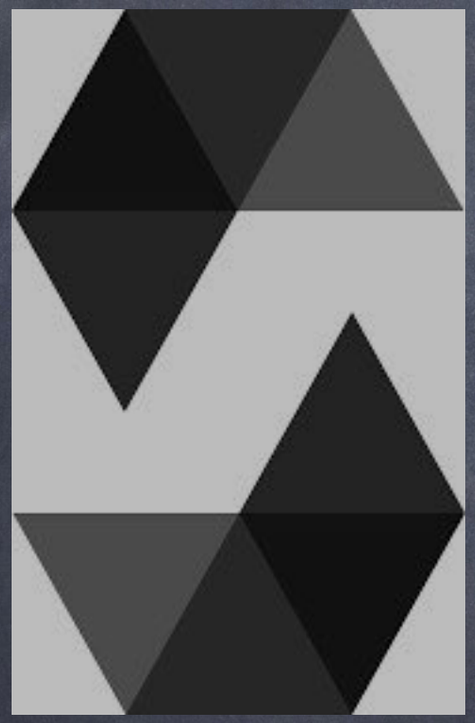
Base -> State Var x
Derived X-> State Var x

Derived -> Any Base
No Visible State Vars

Shadowing -> Error

#107

Overriding Changes



Function Overriding
virtual -> override

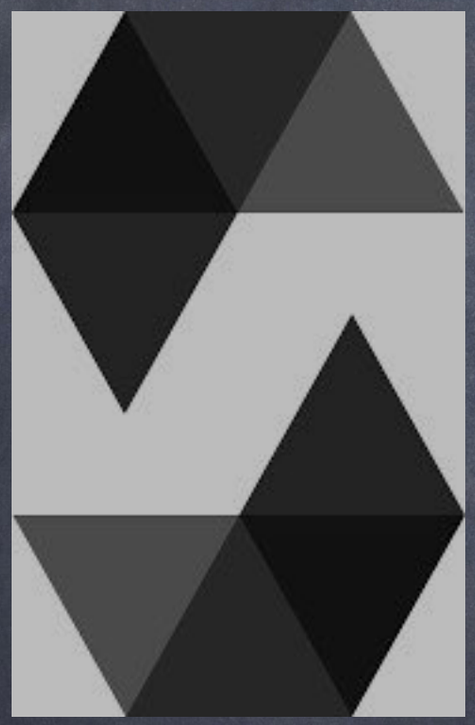
Visibility: external -> public

non payable -> view/pure
view -> pure

payable X-> Any

#108

Virtual Functions



Virtual Function Rules

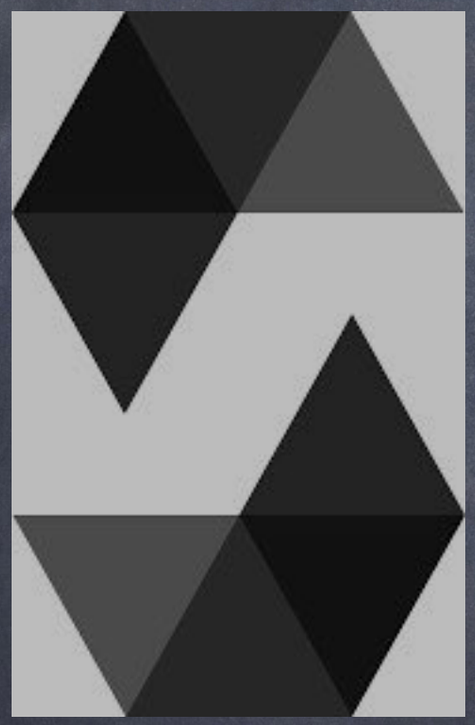
No Implementation -> virtual

interface -> All Functions Considered virtual

private Visibility -> Cannot be virtual

#109

State Variables



Public State Variables

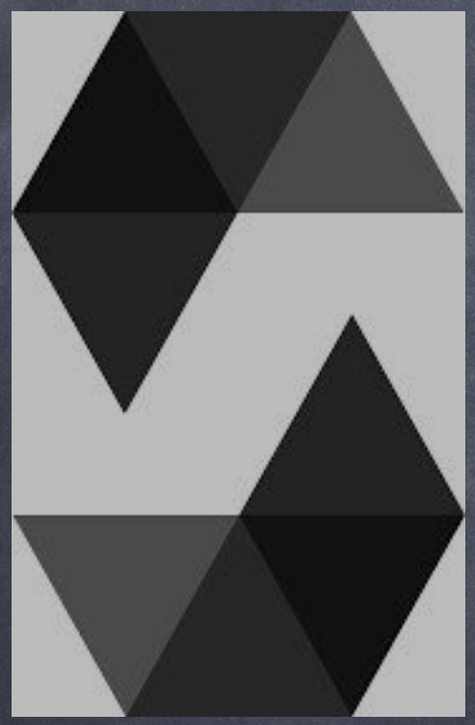
Automatic Getters

external Fns -> Params & Return Types

Getters -> Override external Fns
Override X-> Getters

#110

Function Modifiers



Modifier Overriding

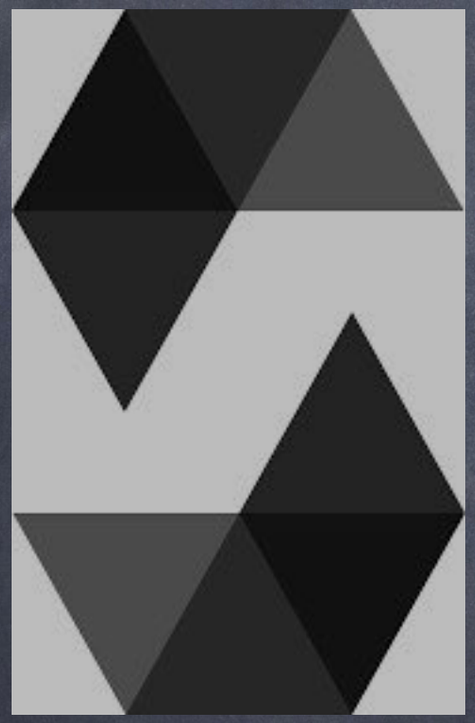
Overriding Possible
Similar to Functions

Overridden Modifier ->
virtual Keyword

Overriding Modifier ->
override Keyword

#111

Base Constructor



Base -> Derived Constructors

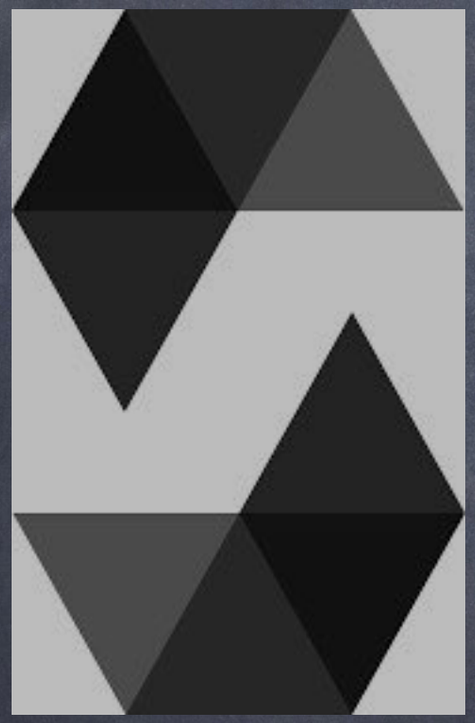
Constructors Called -> Linearization Rules

Constructor Arguments

Inheritance List or Derived Constructor

#112

Name Collision



Pairs -> Same Name
Inheritance -> Errors

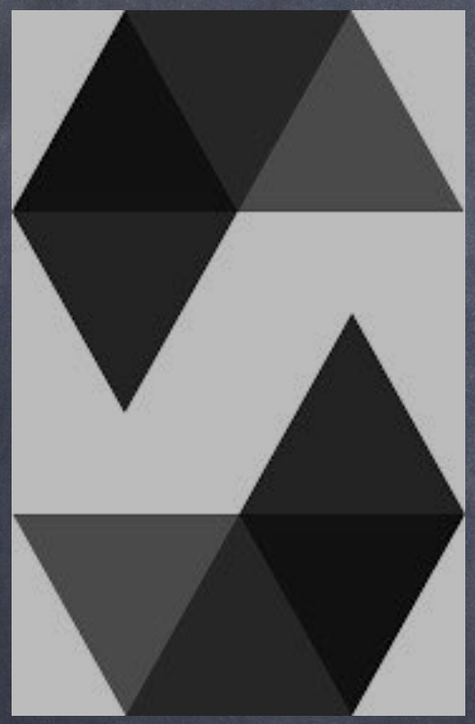
Function & Modifier

Function & Event

Event & Modifier

#113

Library Restrictions



No State Variables
X-> Inherit or Inherited

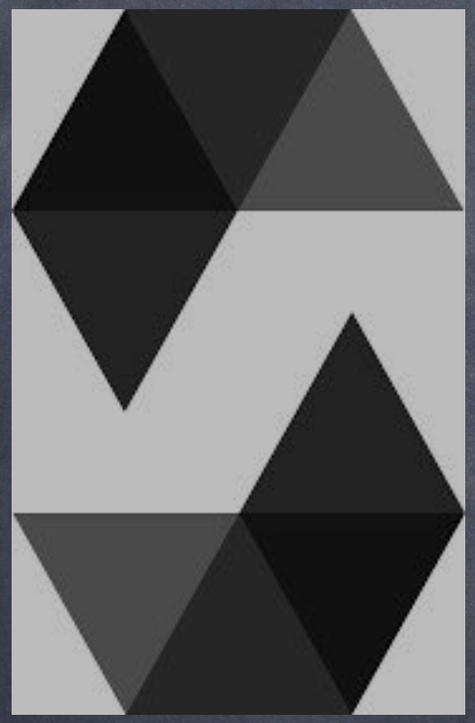
X-> Receive Ether
X-> Be Destroyed

Calling Contract -> Only State Vars Supplied

Directly Vs delegatecall
pure or view Functions

#114

EVM Storage



Storage -> Key/Value Store

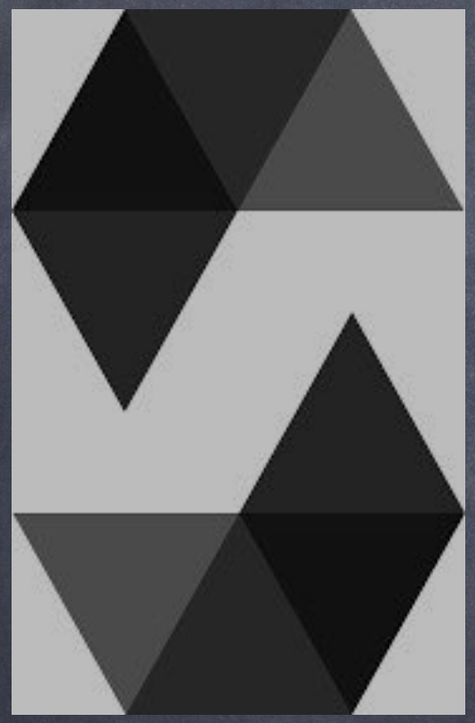
Key -> Value
256-bit -> 256-bit

Storage Access
SLOAD & SSTORE

Storage Location
Zero Initialized

#115

Storage Layout



Storage Variables
Storage Slots

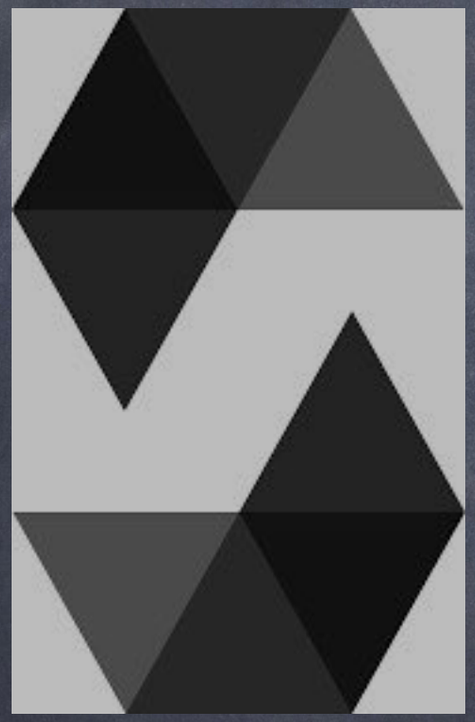
Multiple Vars
Compact -> Same Slot

State Var Order
Stored Contiguously

Exceptions: Dyn Arrays & Mappings

#116

Storage Packing



State Variables
Type -> Size in Bytes

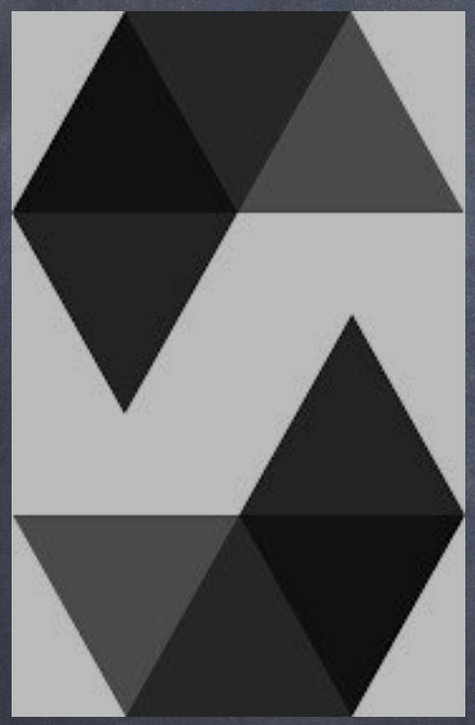
Contiguous & Size Fits
Same Storage Slot

Contiguous & Not Fit
Next Storage Slot

First Item -> Lower-order
Aligned

#117

Structs & Arrays



Structs & Arrays
Storage Rules

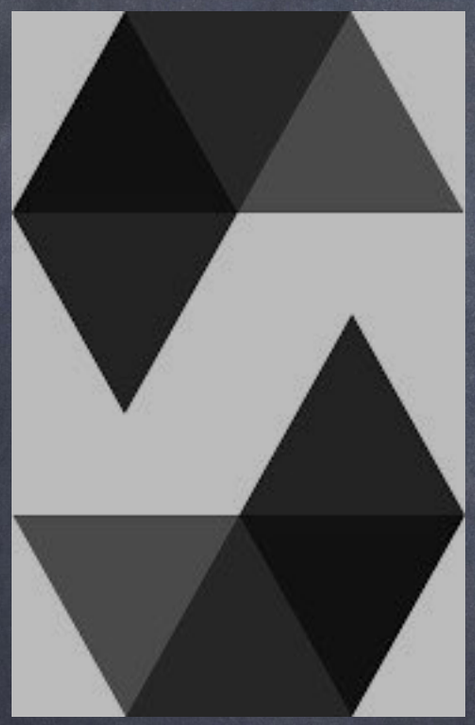
Structs & Arrays -> Start
New Storage Slot

Following Items -> Start
New Storage Slot

Elements -> Contiguous
Individual Values

#118

Inheritance



Contract Inheritance -> Storage Layout

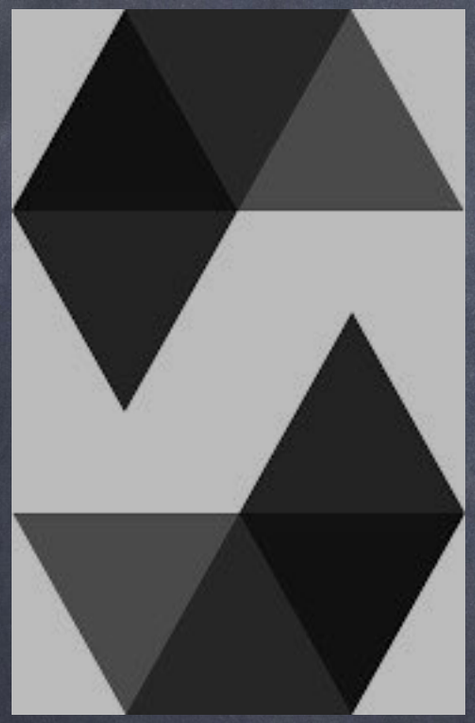
Base(s) -> Derived State Vars Layout

C3-Linearized Order
Most Base-ward -> Derived

State Vars -> Diff Contracts
Rules -> Same Storage Slot

#119

Layout & Types



Reduced-size Types
Combined Slot

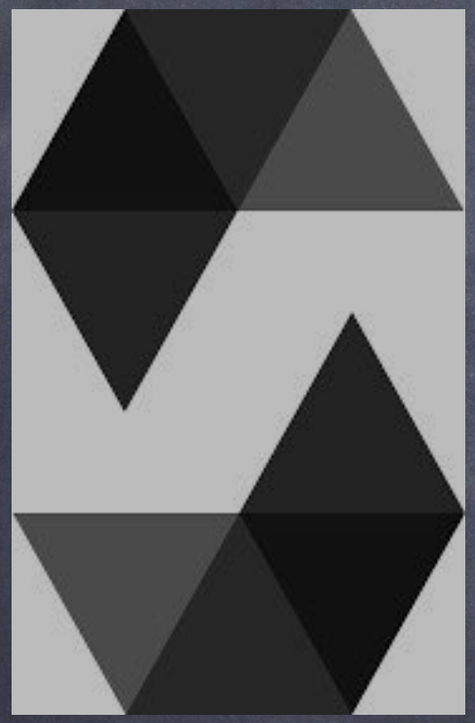
Combine Multiple R/Ws
Same Operation

Multi-value Storage Slot

Values: Some Vs All
Read+Combine+Write

#120

Layout & Ordering



State Vars Ordering
Declaration -> Packing

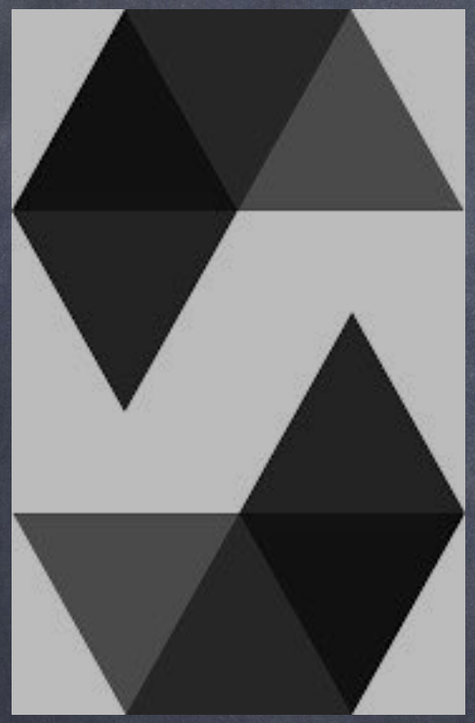
Packing -> Gas Costs
SLOADs & SSTOREs

uint128, uint128, uint256
Two Storage Slots

uint128, uint256, uint128
Three Storage Slots

#121

Mappings & Dyn Arrays



Storage Layout
Unpredictable Size

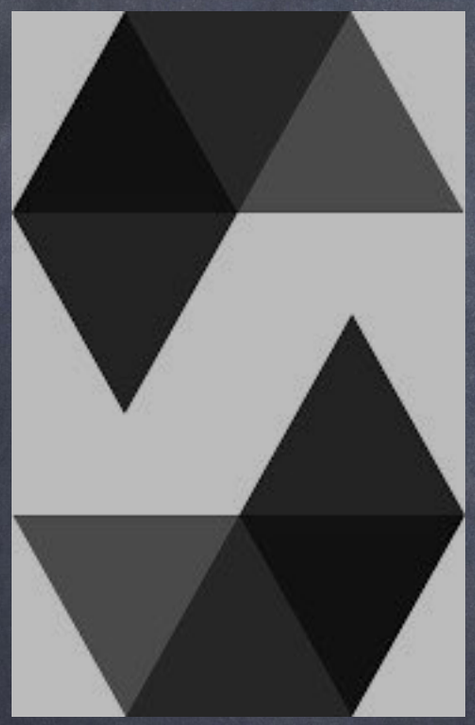
Cannot be Stored
In-between Others

Occupy -> 32 Bytes
Elements -> Elsewhere

Elements -> Starting Slot
Keccak256 Hash

#122

Dyn Arrays



Array -> Slot p
Stores -> #Array Elements

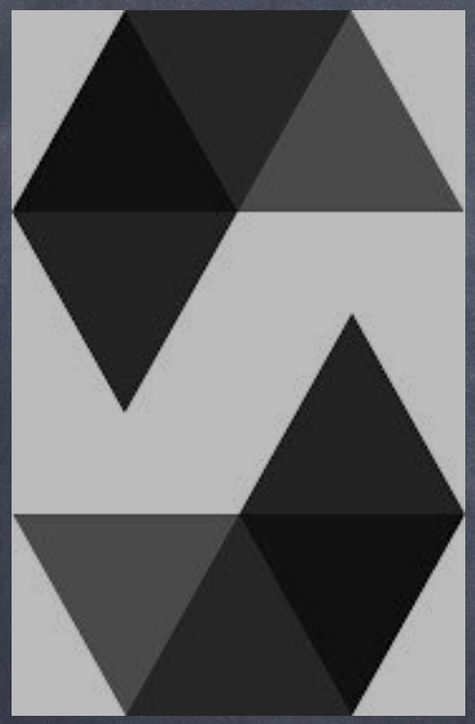
Elements -> keccak256(p)

Stored Contiguously
Share Slots if Possible

Dyn Arrays -> Dyn Arrays
Recursively Apply

#123

Mappings



Mapping -> Slot p
Stores -> Nothing

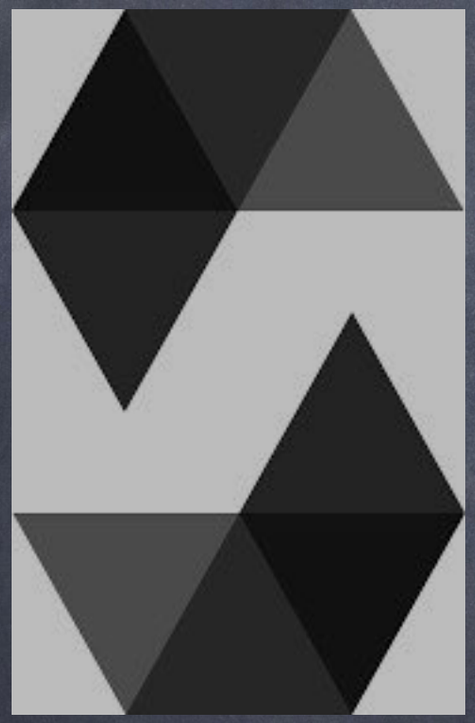
Mapping[Key k] ->
keccak256(h(k).p)

. -> Concatenation
h -> Type-specific Function

Value -> Padded 32B
string/byte -> keccak256

#124

bytes & string



Storage Layout
Similar to Arrays

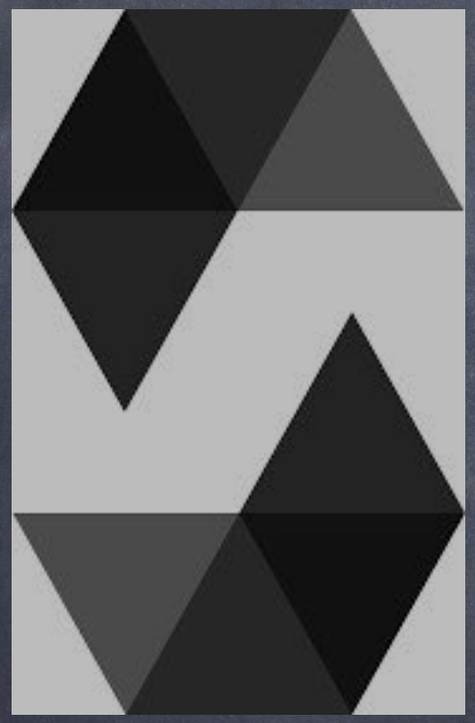
Short Values
Len+Elements -> Same Slot

<= 31B -> Length*2
>= 32B -> Length*2 + 1

Lowest Bit -> 0 or 1
Array Type -> Short or Long

#125

Memory



EVM Memory
Linear Layout

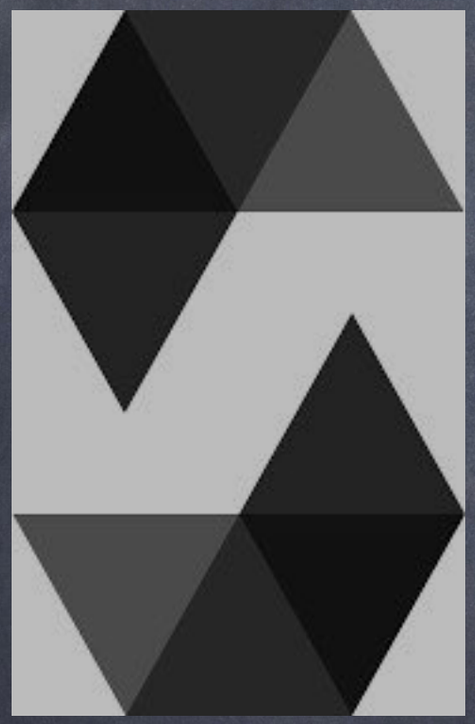
Addressed -> Byte Level

MLOAD/MSTORE/MSTORE8

Zero Initialized

#126

Memory Layout



New Memory Objects
Free Pointer

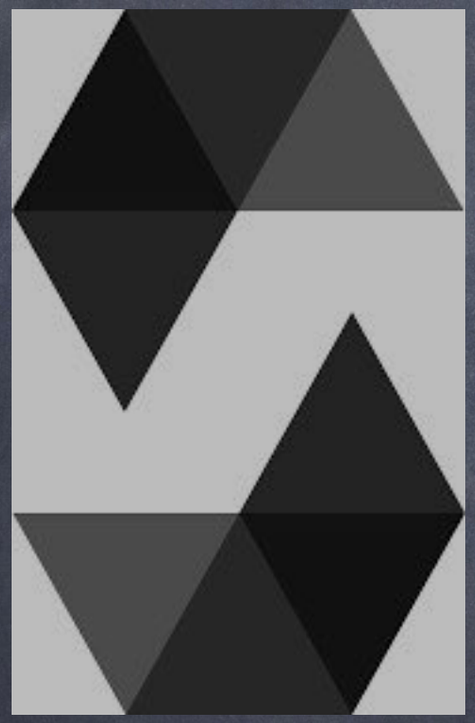
Memory -> Never Freed

Free Memory Pointer ->
Initially 0x80

Security: Memory
Manipulations in Assembly

#127

Reserved Memory



Solidity -> 4 32B Slots

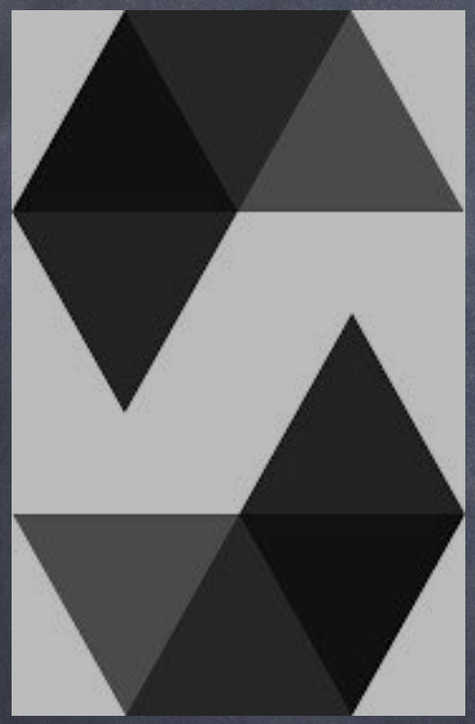
0x00-0x3f (64B)
Hashing -> Scratch Space

0x40-0x5f (32B)
Free Memory Pointer

0x60-0x7f (32B)
Zero Slot

#128

Memory
Arrays



Elements -> 32B Multiples

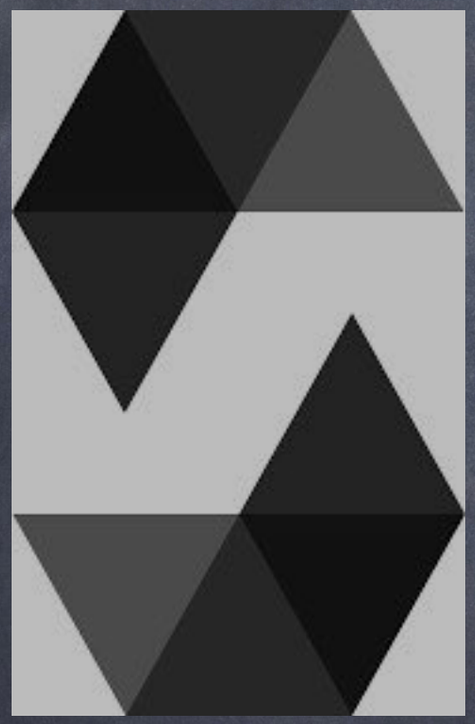
Even byte[]
Not bytes & string

Multi-dimensional
Pointers -> Arrays

Dynamic Arrays
Length+Elements

#129

Free Memory Pointer



Position 0x40

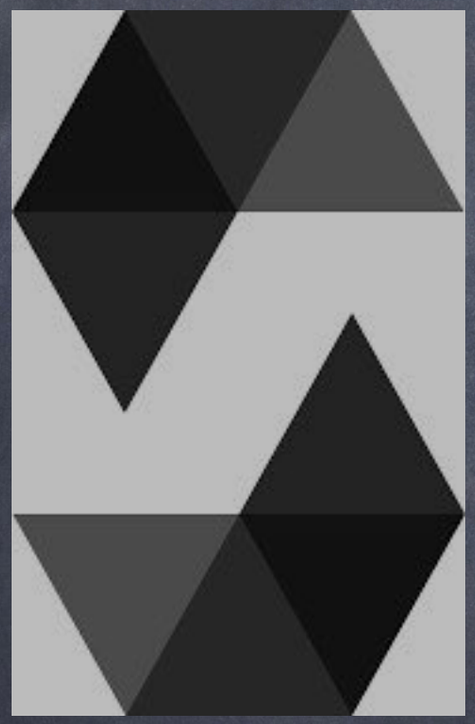
Initial Value -> 0x80
Beyond Reserved Slots

Ptr Points to ->
Allocatable Memory

Memory Allocation ->
Update Free Memory Ptr

#130

Zeroed Memory



Memory Not Used -> No Guarantee

Zeroed Contents -> Cannot Assume

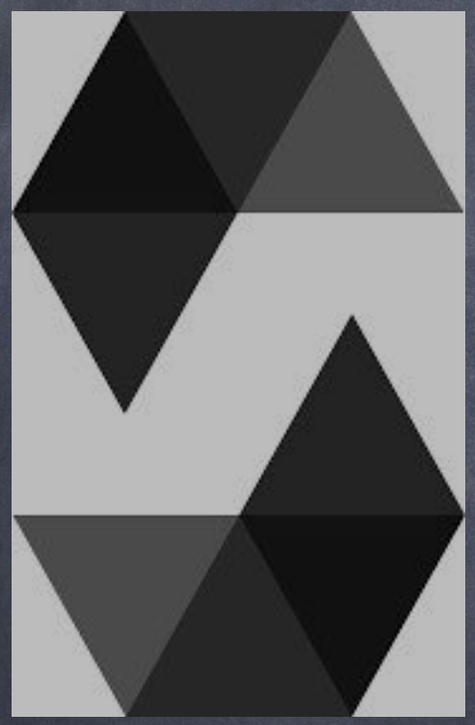
Release/Free Memory
No Built-in Mechanism

Security: Memory Manipulations in Assembly

#131

Reserved Keywords

Keywords -> Reserved for Future



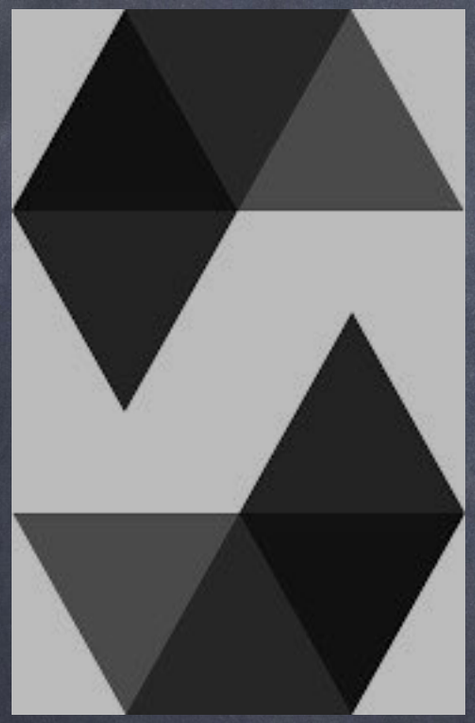
Current Syntax -> No
Future Syntax -> Maybe

E.g.: after, alias, apply, auto, case, null etc.

E.g.: unchecked in v0.8.0

#132

Inline Assembly



Access EVM
Low-level Features

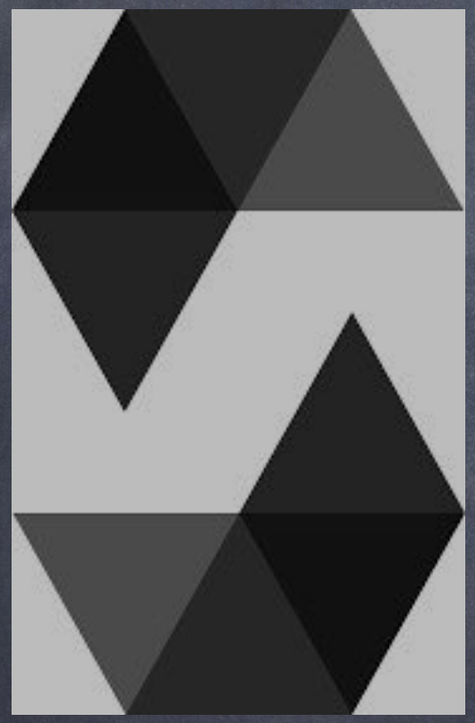
Bypasses Safety Features
E.g.: Type Safety

Language -> Yul

Assembly Block
assembly {...}

#133

Assembly Access



External Variables, Functions & Libraries

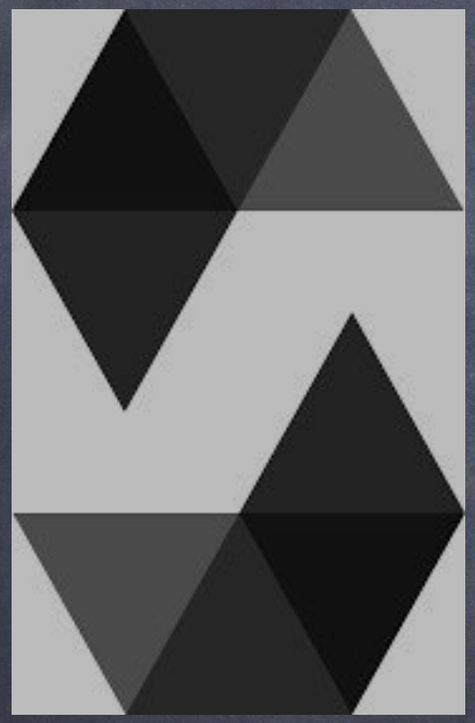
Local Vars -> Value Type
Value Vs Addr

Storage Vars -> Slots
*.slot & *.offset

Assignments Possible
Rules & Restrictions

#134

Yul
Syntax



Literals & Calls

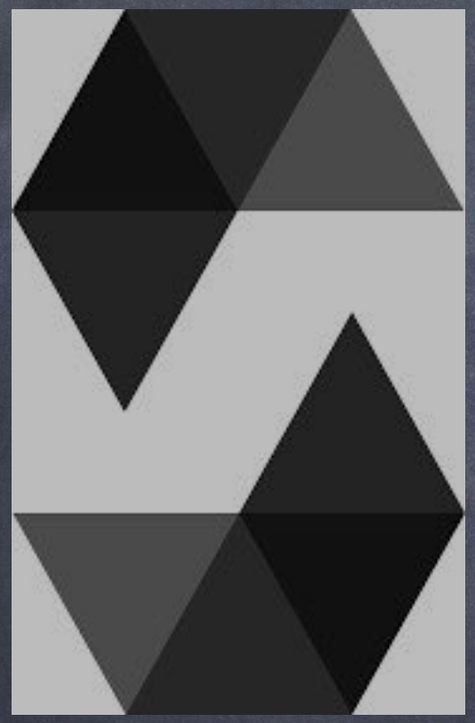
Variable Declarations
Assignments

Scoping Blocks
if/switch/for

Function Definitions

#135

solc 0.6.0
Breaking



Breaking Semantic
Changes

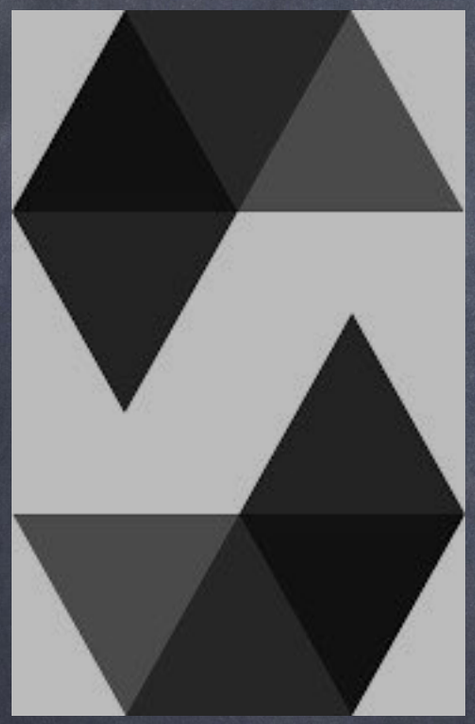
Existing Code
Changes Behavior

Exp Result -> Base Type

Not Smallest Type
Both Base & Exponent

#136

solc 0.6.0
Explicitness



virtual -> override
Array Length -> Read-only

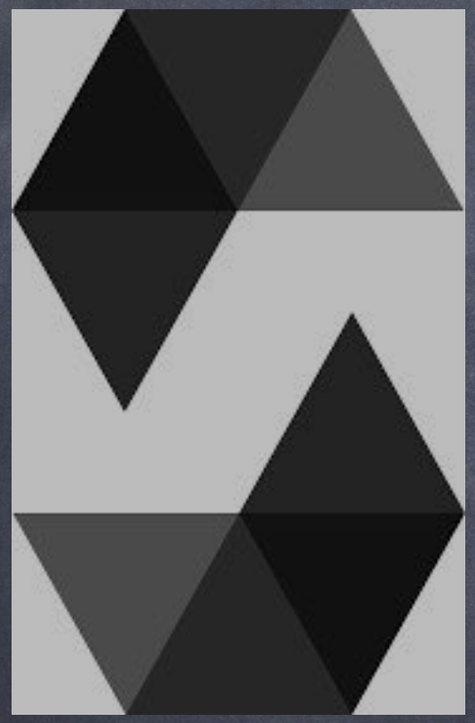
abstract Keyword
Libraries -> All Functions

Assembly Variables
Restrictions

State Variable Shadowing
Not Allowed

#137

solc 0.6.0
Changes



external Fn X-> address
address Member

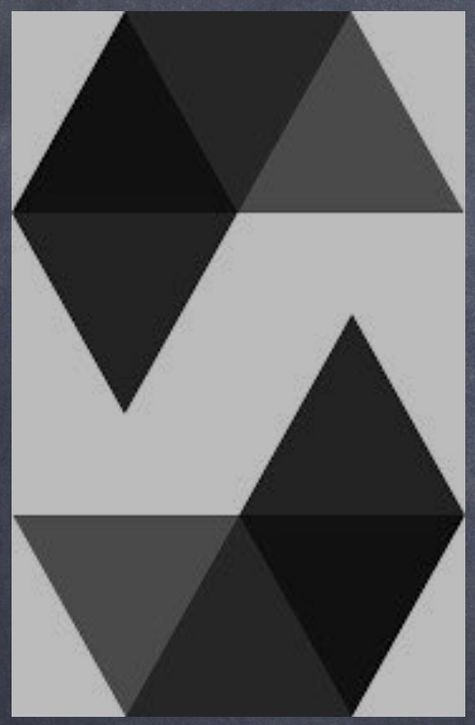
Dyn Storage Arrays
push(x) -> Returns Nothing

Unnamed Function ->
fallback() & receive()

fallback() Vs receive()

#138

solc 0.6.0
New Features



try/catch
Failed External Calls

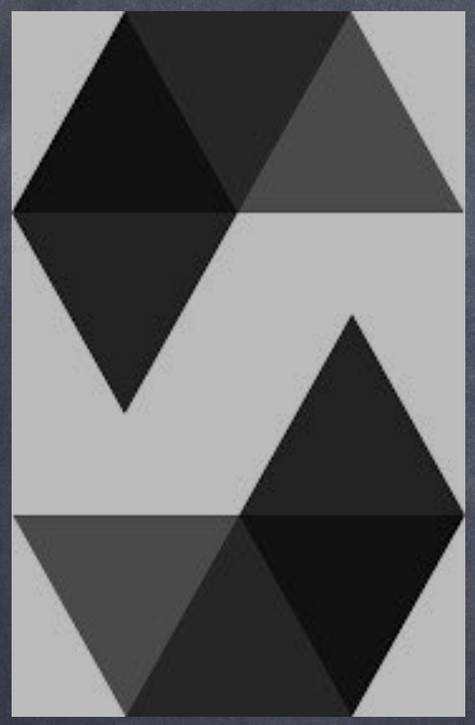
struct/enum -> File-level

Array Slices -> calldata
Natspec -> return

Yul -> leave
payable(x)

#139

solc 0.7.0 Breaking



Breaking Semantic Changes

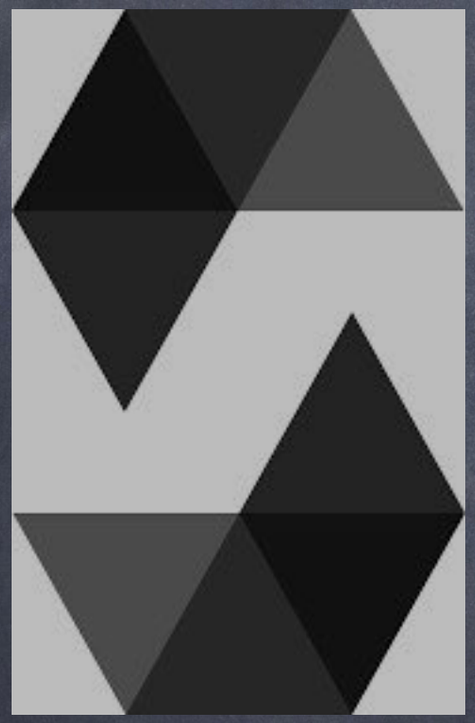
Exponentiation & Shifts Literals by Non-literals

Earlier -> Type of shift/ exponent

Now -> uint256 or int256

#140

solc 0.7.0
Changes



Calls -> gas & value
now -> block.timestamp

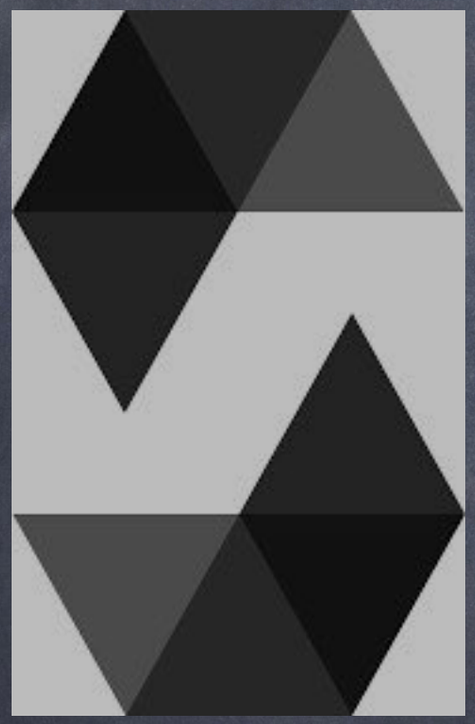
Natspec -> Vars
gwei -> Keyword

String Literals
Unicode

Fn State Mutability
Assembly Changes

#141

solc 0.7.0
Removed



Struct/Array -> Mapping
Constructor Visibility

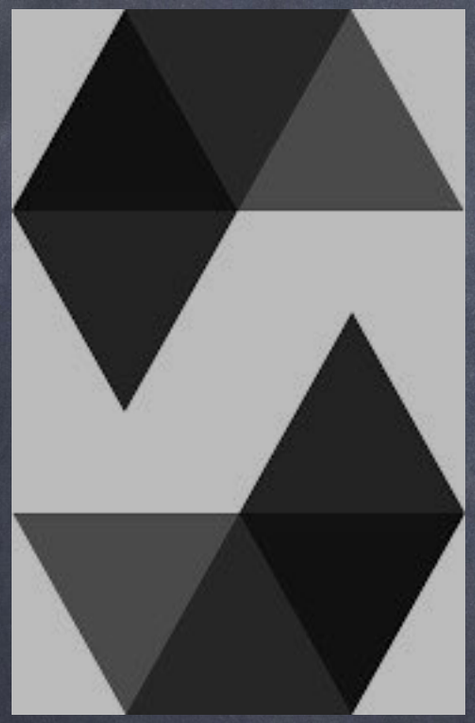
Virtual Lib Fns
Events -> Same Name/Params

using A for B
Contract X-> Derived

Shifts by Signed Types
finney/szabo/var

#142

solc 0.8.0
Breaking



Checked Arithmetic
unchecked {...}

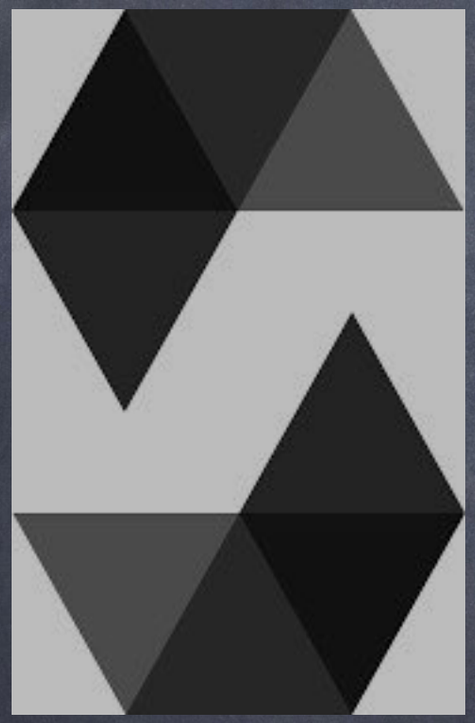
Default ABI coder v2
a**(b**c) Vs (a**b)**c

Revert Vs Invalid Opcodes

Storage byte Arrays
Type byte Removed

#143

solc 0.8.0
Restrictions



Explicit Conversions
Addr Literals -> address

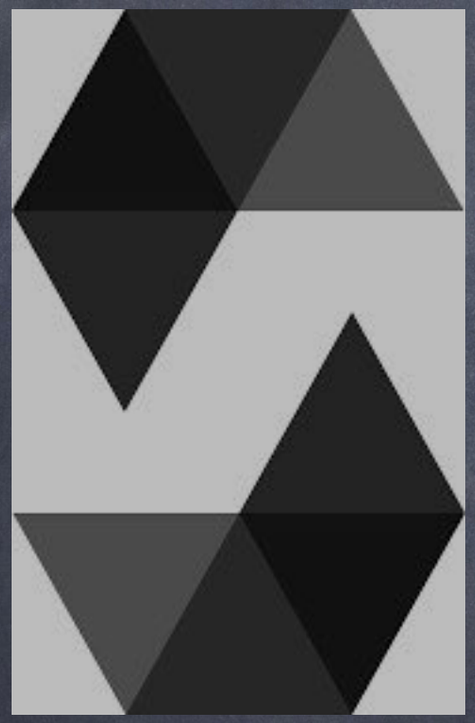
Fn Call Options -> Once
logo-log4 -> Removed

enum -> 256/uint8
X-> this, super, _

tx.origin/msg.sender
chainID -> view

#144

Zero-address Check



address(0) X-> Private Key

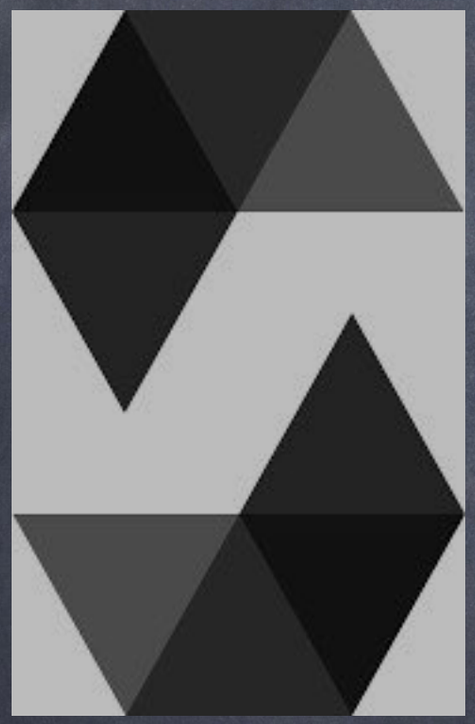
Transfer Ether/Token -> Burn

Access Control
Special Consideration

Zero-address Checks
Addr Parameters

#145

tx.origin
Check



Ethereum Accounts
EOA Vs Contract

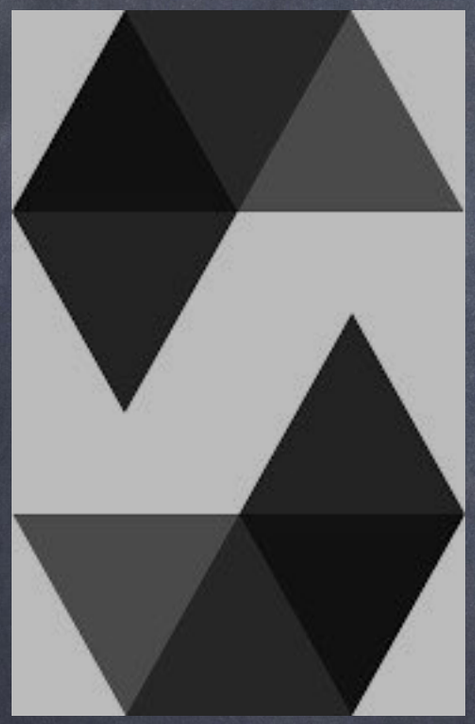
Tx Origin -> Only EOA

msg.sender == tx.origin

Sender -> Contract or Not

#146

Arithmetic Check



Arithmetic Overflow/ Underflow -> Wrapping

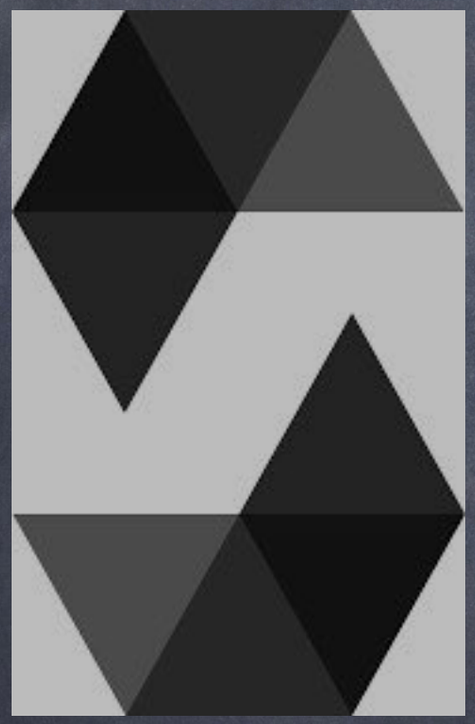
Balances & Accounting Critical Vulnerabilities

solc < 0.8.0
SafeMath Library

solc >= 0.8.0
Default Checks

#147

OZ
Libraries



OpenZeppelin
Smart Contract Libraries

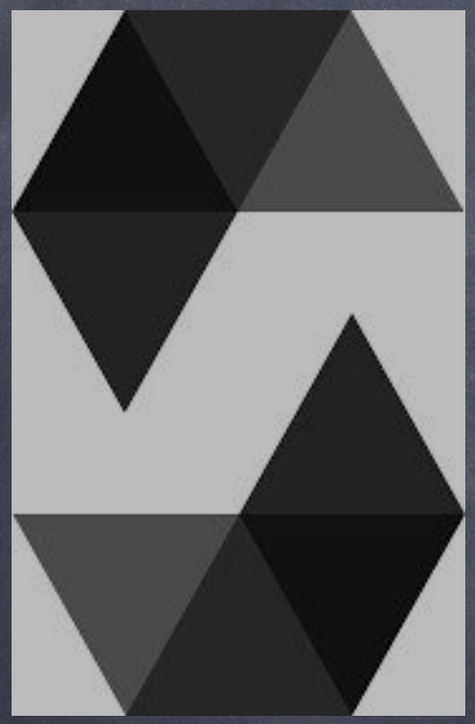
Widely Used
Time-tested & Optimized

SafeMath Library

Token Standards, Security,
Proxy, Utils

#148

OZ
ERC20



ERC20 Token Standard

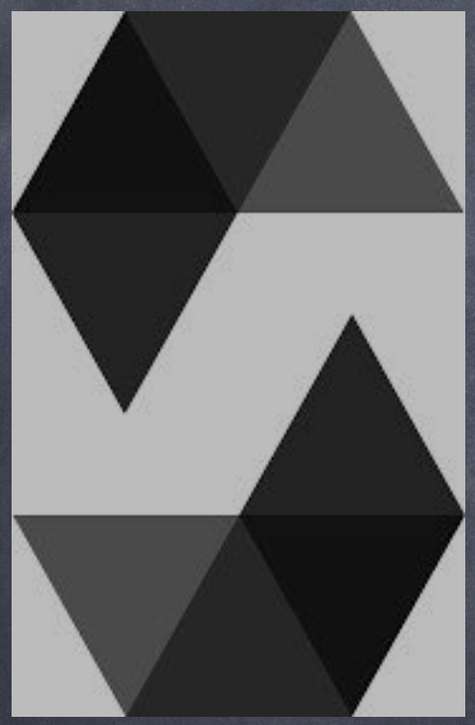
All Required Functions

Increase/Decrease
Allowance

Extensions/Presets/Utils

#149

OZ ERC20 Util
SafeERC20



transfer/transferFrom/
approve/increaseAllowance/
decreaseAllowance

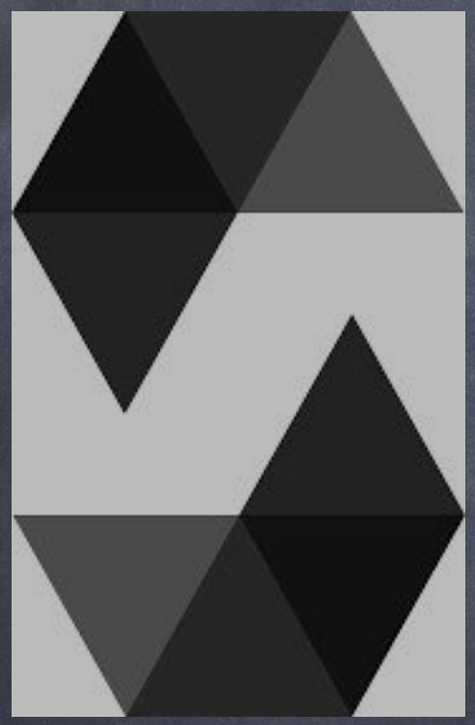
bool Vs revert Vs Nothing

safe* Wrappers -> Revert
on Failure

using SafeERC20 for
IERC20

#150

OZ ERC20 Util
TokenTimelock



Token Holder Contract

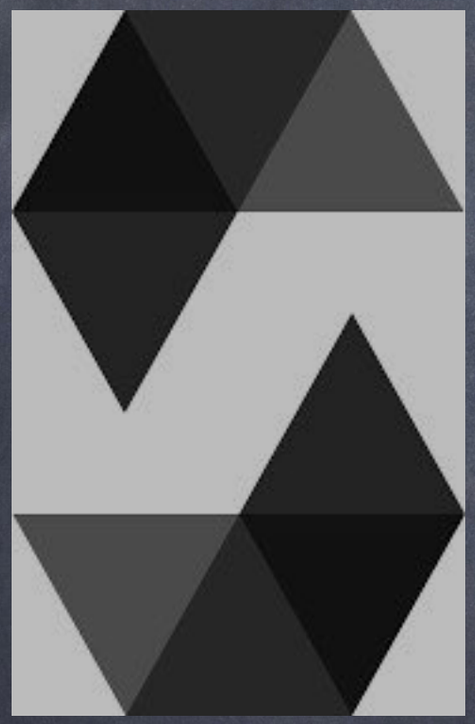
Token Beneficiary ->
Timed Release

Token Vesting -> Advisors/
Team etc.

token, beneficiary
releaseTime -> release()

#151

OZ
ERC721



ERC721 Token Standard
NFTs

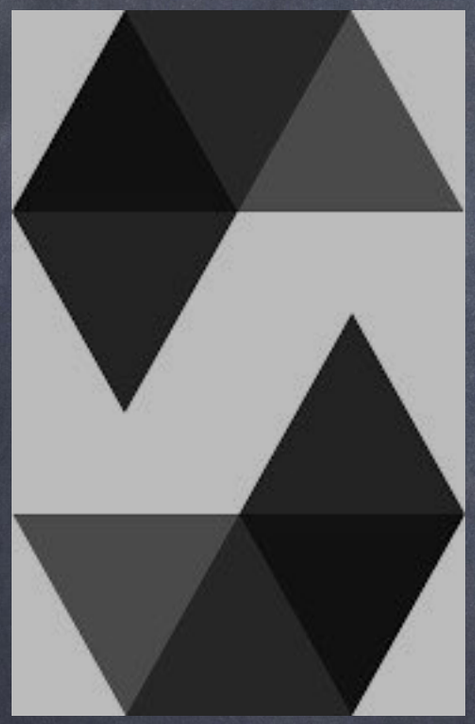
All Required Functions

Diffs with ERC20
Transfers/Approvals/Operators

Extensions/Presets/Utils

#152

OZ
ERC777



ERC777 Token Standard
ERC20 Improvements

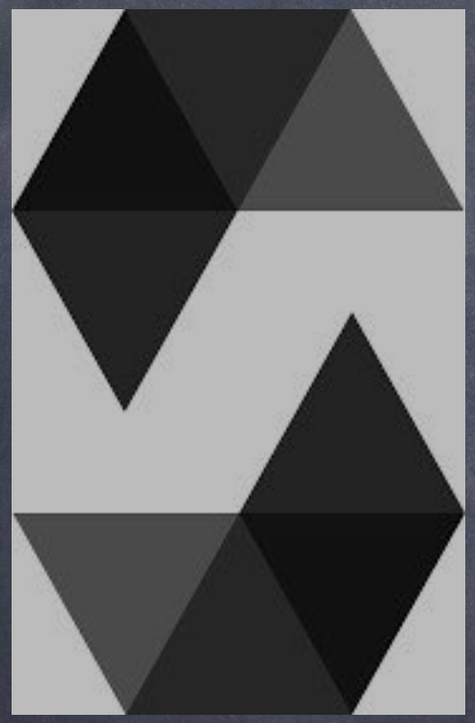
Key Feature: Hooks
tokensToSend/tokensReceived

No approve/transferFrom
No Stuck Tokens

18 Decimals, Operators,
send()

#153

OZ
ERC1155



ERC1155 Token Standard
Fungibility-agnostic

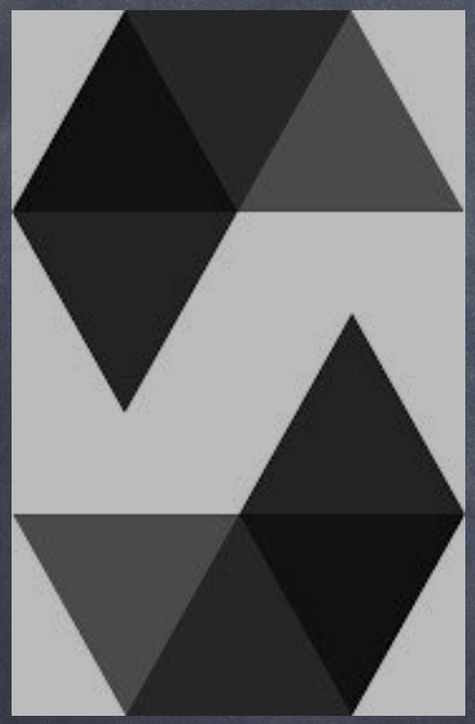
Single Contract
Multiple Tokens

Convenient & Efficient
Single Tx -> Multiple Tokens

balanceOfBatch()
safeBatchTransferFrom()

#154

OZ
Ownable



Basic Access Control
Contract -> Owner

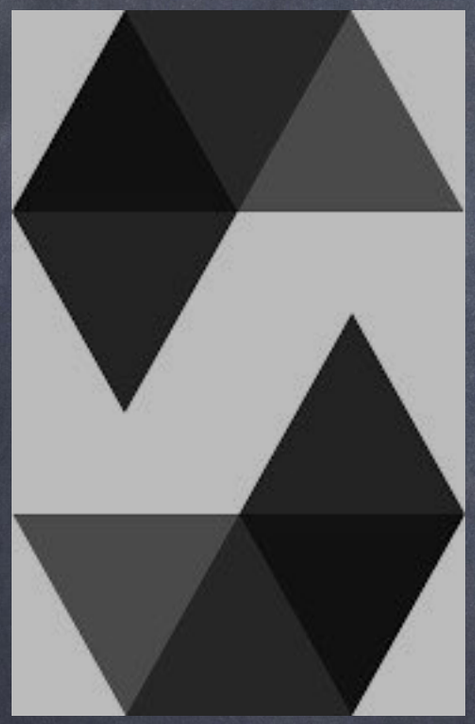
Default Owner -> Contract
Deployer

Exclusive Owner Access ->
Special Functions

Modifier onlyOwner
transferOwnership()

#155

OZ
AccessControl



Generalized RBAC
Role-based Access Control

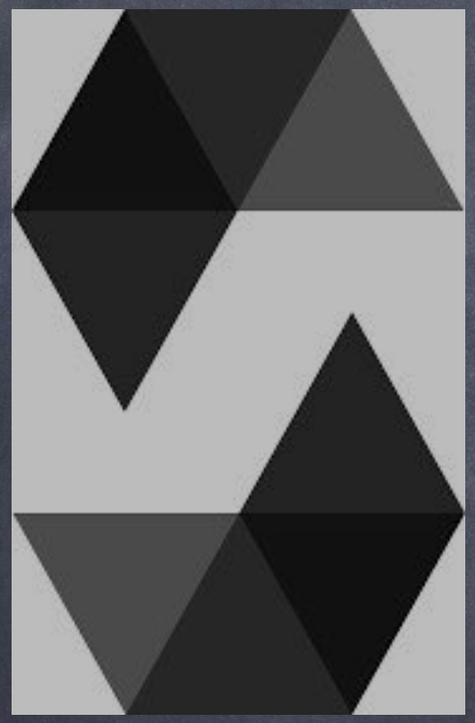
Roles -> Permission Set
onlyRole -> Restrict Access

grantRole & revokeRole
Role -> RoleAdmin

Ownable -> Simple
AccessControl -> Flexible

#156

OZ Pausable



Guarded Launch
Emergency -> Remediate

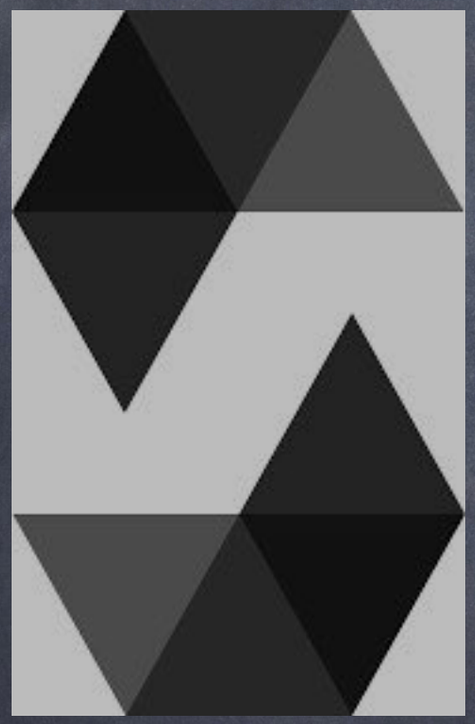
pause() & unpause()
Authorized Access

Modifiers: whenPaused & whenNotPaused

Vulnerability or Exploit
Circuit-breaker

#157

OZ
ReentrancyGuard



Reentrancy Vulnerability
Unique & Dangerous

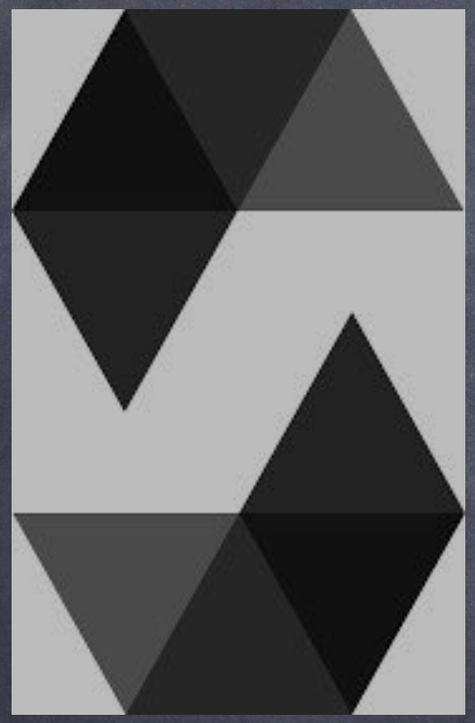
External & Untrusted
Calls -> Reenter & Exploit

Modifier: nonReentrant

Prevents Reentrancy
Best Practice

#158

OZ
PullPayment



Payment: Pull Vs Push
Avoid Reentrancy Attacks

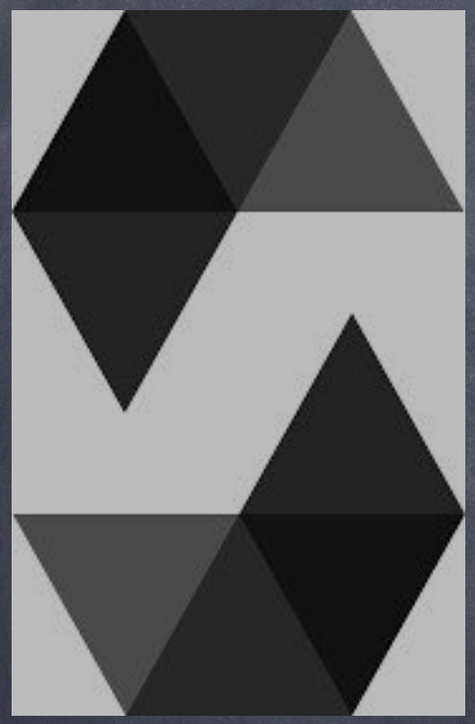
Paying Contract
No Calls -> Receiver Fns

Receiver Account
Withdraws Payment

Prevents Reentrancy
Best Practice

#159

OZ
Address



Address-related Functions

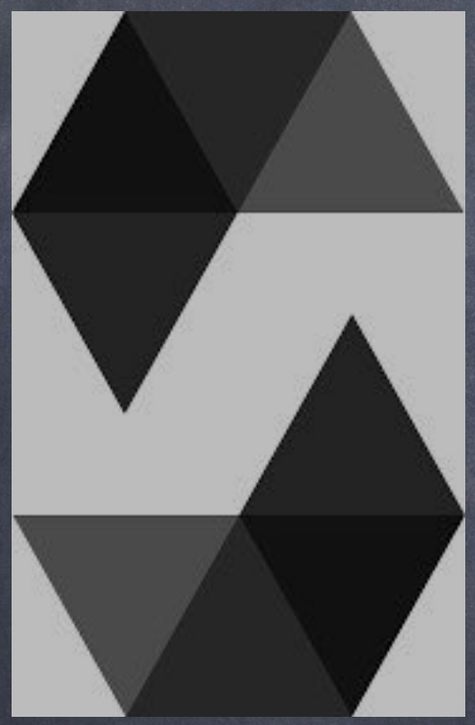
isContract()
sendValue()

functionCall()
functionCallWithValue()

functionStaticCall()
functionDelegateCall()

#160

02
Arrays



Array-related Functions

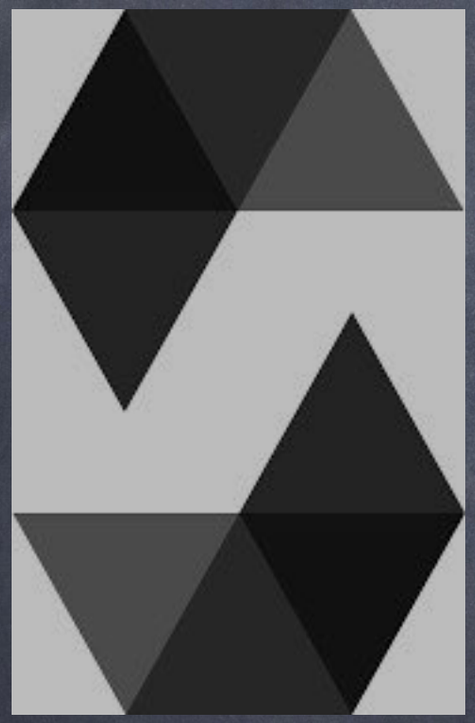
findUpperBound()
Params: Array & Element

Array: Sorted Ascending
No Repeats

Value >= Element -> Index
None -> Length

#161

OZ
Context



Current Execution Context

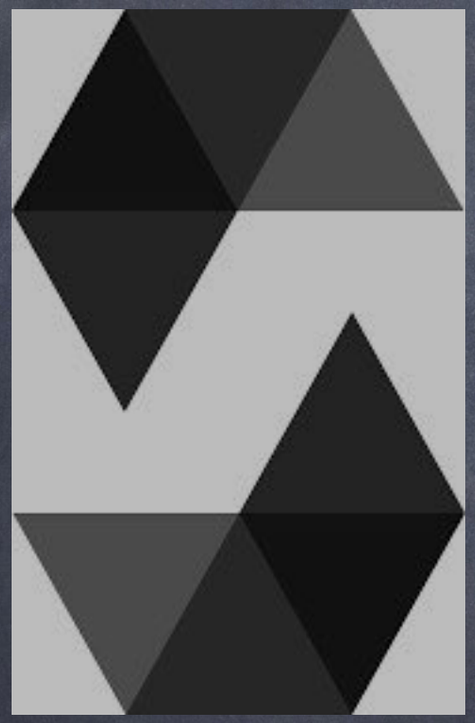
msg.sender & msg.data

Meta-transactions
Sender X= User

_msgSender() & _msgData()

#162

OZ
Counters



Counters
Increment & Decrement

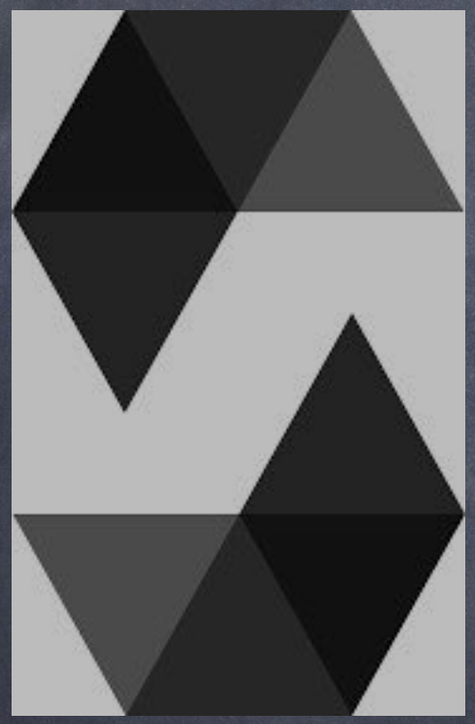
Mapping Elements, ERC721
IDs, Request IDs

current() -> Value
reset() -> 0

increment()
decrement()

#163

OZ
Create2



CREATE2 EVM Opcode
Easier & Safer

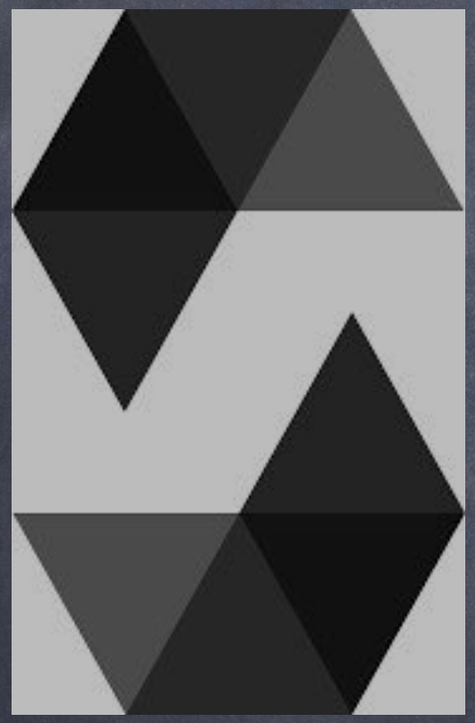
deploy (uint256 amount, bytes32 salt, bytes bytecode)

computeAddress (bytes32 salt, a bytecodeHash)

computeAddress (bytes32 salt, bytes32 bytecodeHash, address deployer) → address

#164

OZ
Multicall



Batch Calls
Single External Call

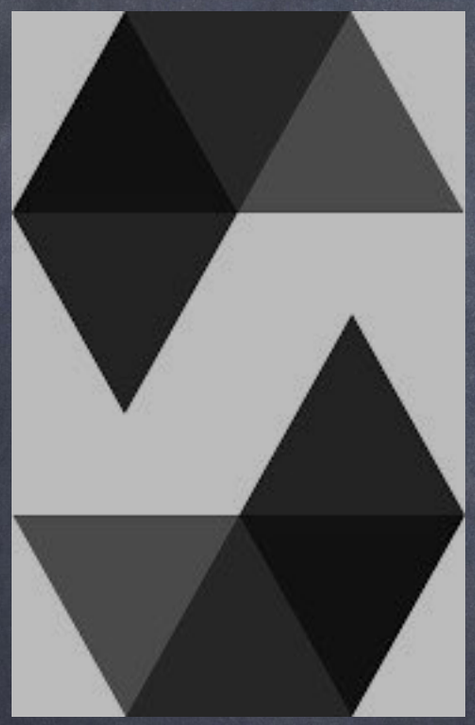
multicall (bytes[] calldata
data) external → bytes[]

Receives + Executes
Function Calls -> Batch

One Tx, Same Block
Less Overhead -> Gas Efficient

#165

OZ
String



String Operations

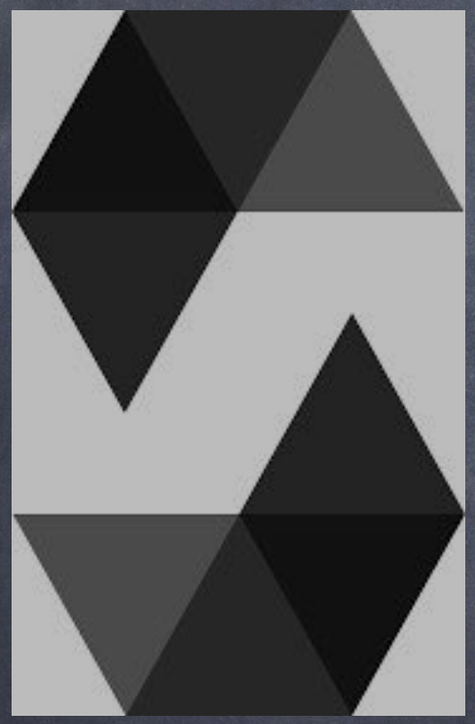
toString(value)
uint256 -> ASCII String Dec

toHexString(value)
uint256 -> ASCII String Hex

toHexString(value, length)
ASCII String Hex, Fixed Length

#166

OZ
ECDSA



ECDSA Signatures

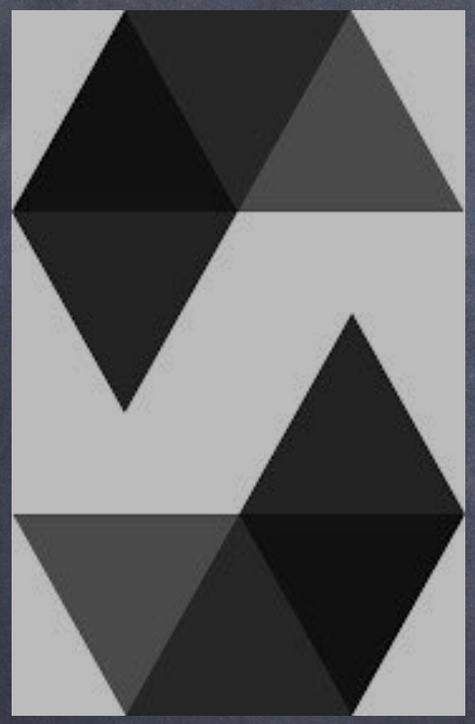
Signature -> (v, r, s)
v=1B, r=32B, s=32B

recover(msgHash, signature)
-> signerAddress

EVM ecrecover -> Malleable
OZ ECDSA -> v=27/28, s=lower

#167

OZ
MerkleProof



Merkle Tree Proof
Verification

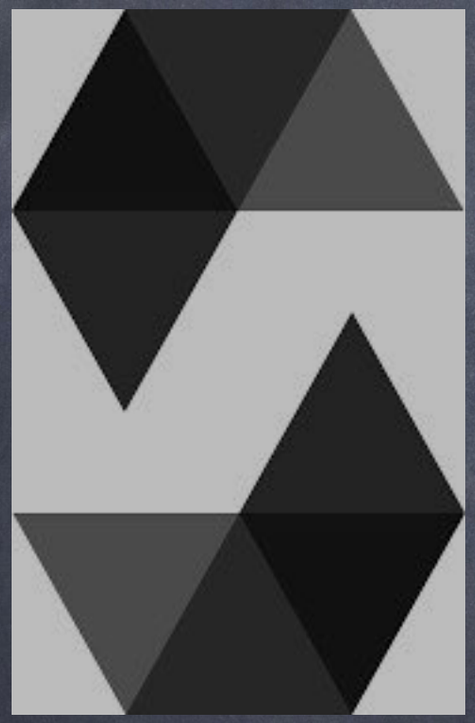
Merkle Tree
Leaf -> Root

verify (proof, root, leaf) -
> bool

Proof -> Sibling Hashes
Branches -> Leaf to Root

#168

OZ SignatureChecker



ECDSA & ERC-1271 Signatures

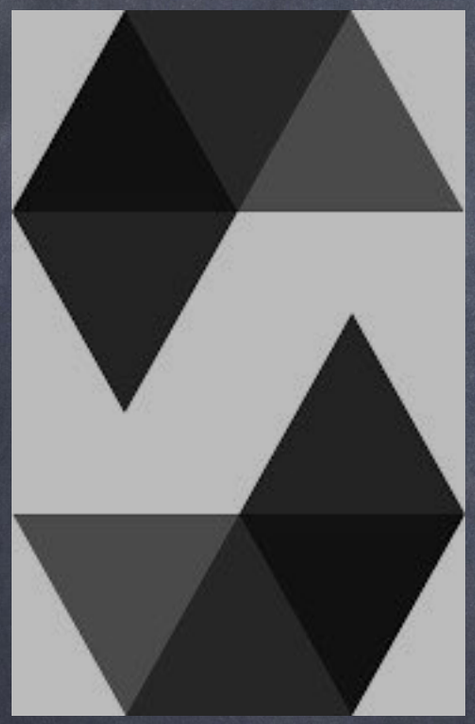
ECDSA -> EOA Signatures

ERC-1271 -> Contract Signatures

Smart Contract Wallets & Other Applications

#169

OZ
EIP-712



Hashing & Signing
Typed Structured Data

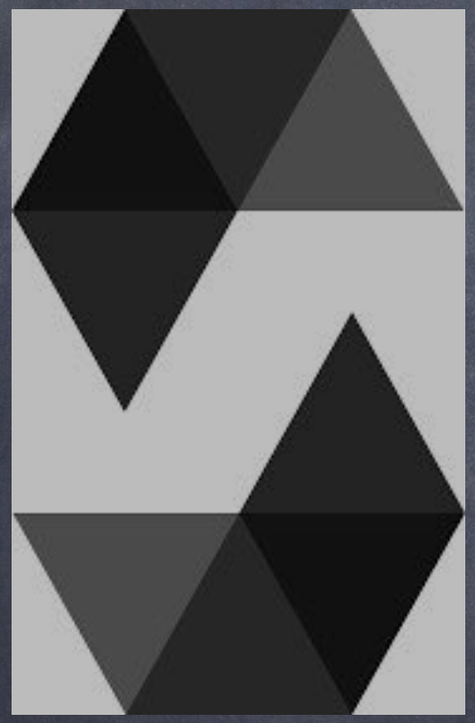
EIP-712 Domain Separator

block.chainID &
address(this)

Prevent Replay Attacks

#170

OZ
Escrow



Escrow Funds for Payee
Ownable

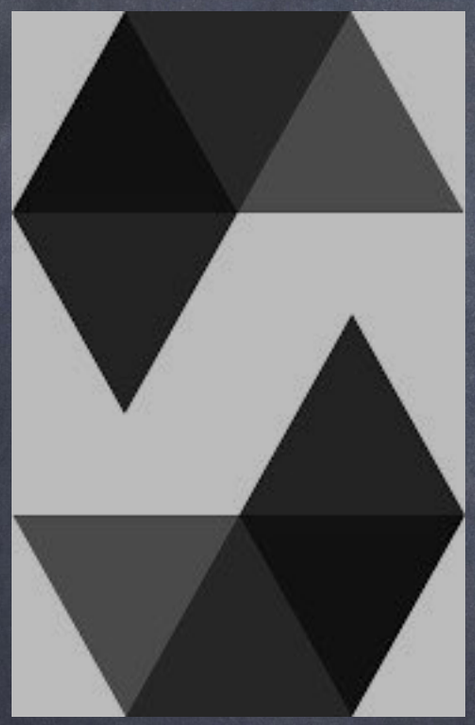
depositsOf (payee) ->
uint256

deposit (payee)
onlyOwner

withdraw (payee)
onlyOwner

#171

OZ
ConditionalEscrow



Escrow Funds for Payee
Conditional Withdrawal

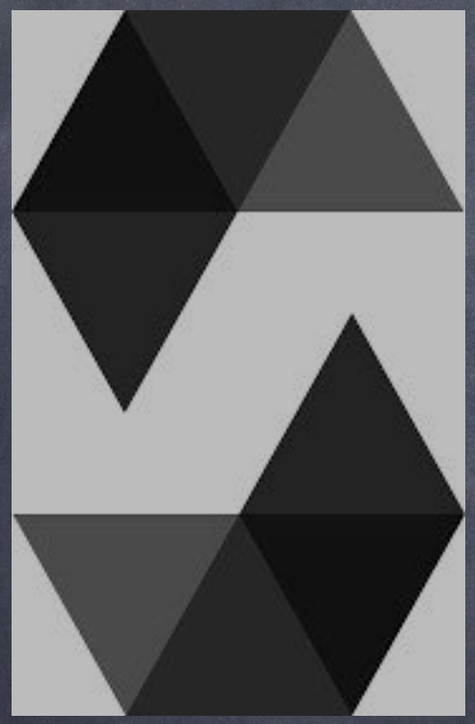
withdrawalAllowed() ->
Check Condition

withdraw (payee)
public Vs onlyOwner

require(withdrawalAllowed
(payee)) -> withdraw

#172

OZ
RefundEscrow



Escrow -> Beneficiary
Multiple Depositors

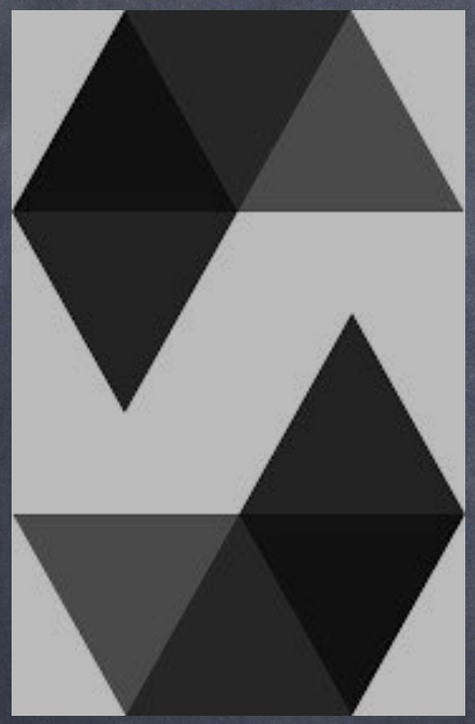
State -> Active/Refunding/
Closed

Active -> Deposits
Refunding -> Refunds

Closed -> Beneficiary
Withdrawals

#173

OZ
ERC-165



Contract -> Interface
Determine Support

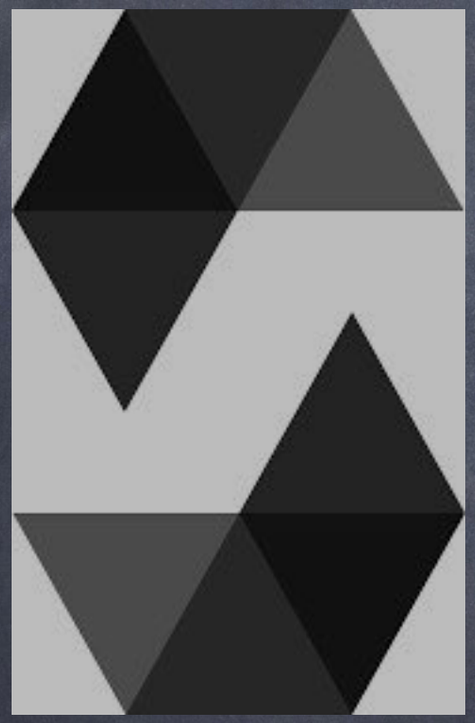
Runtime Detection
Lookup Table

Register Interface ->
_registerInterface(bytes4)

supportsInterface(bytes4
interfaceId) -> bool

#174

OZ
Math



Math Utilities
Missing in Solidity

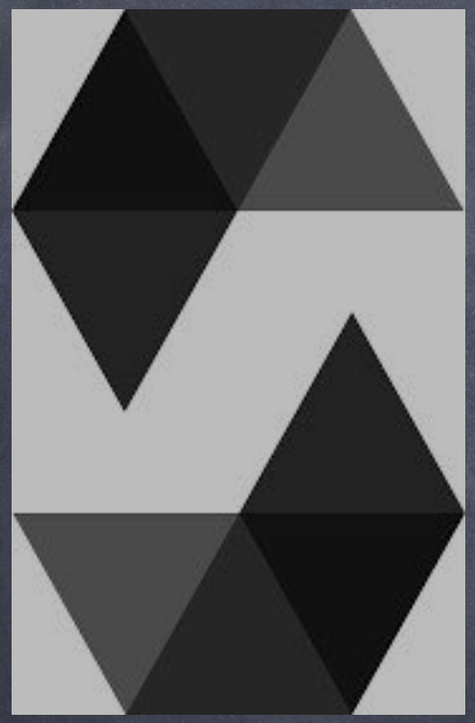
max(uint256 a, uint256 b)

min(uint256 a, uint256 b)

average(uint256 a, uint256 b)

#175

OZ
SafeMath



Math Functions
Safe -> Overflow/Underflow

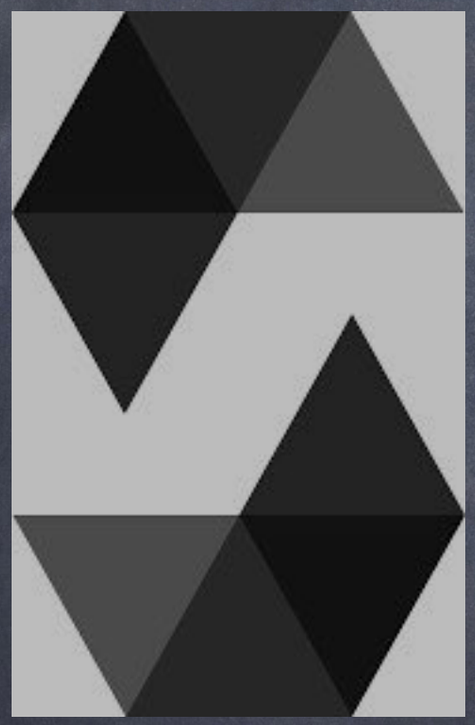
add & sub & mul & div & mod

using SafeMath for uint256

try* -> Flag Vs Revert
Required: solc < 0.8.0

#176

OZ
SignedSafeMath



Math Functions
Safe -> Overflow/Underflow

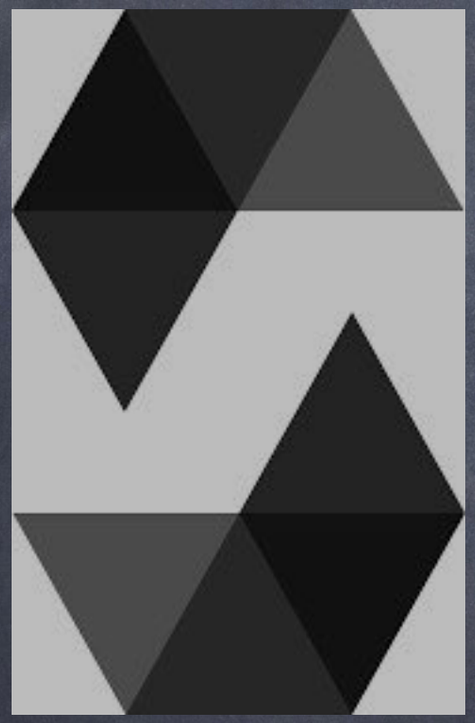
add & sub & mul & div

using SignedSafeMath for int256

Required: solc < 0.8.0

#177

OZ
SafeCast



Downcasting -> Overflow
Safe Downcasting

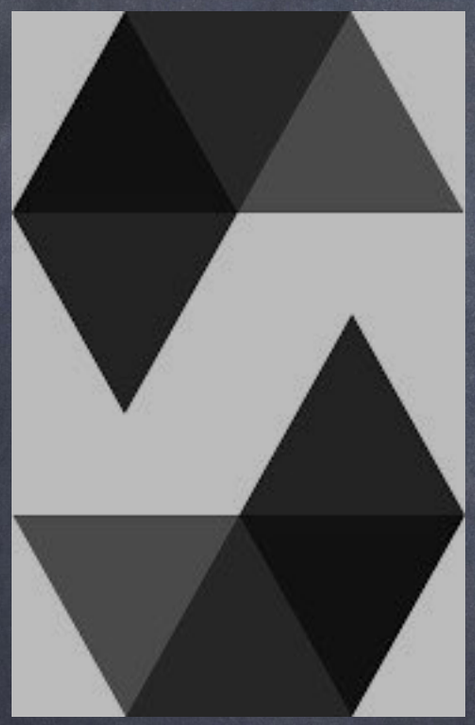
uint256 -> uint224/uint128/
uint96/uint64/uint32/uint16/
uint8

int256 -> int224/int128/uint96/
int64/int32/int16/int8

uint256 -> int256
int256 -> uint256

#178

OZ
EnumerableMap



Mapping -> Enumerable

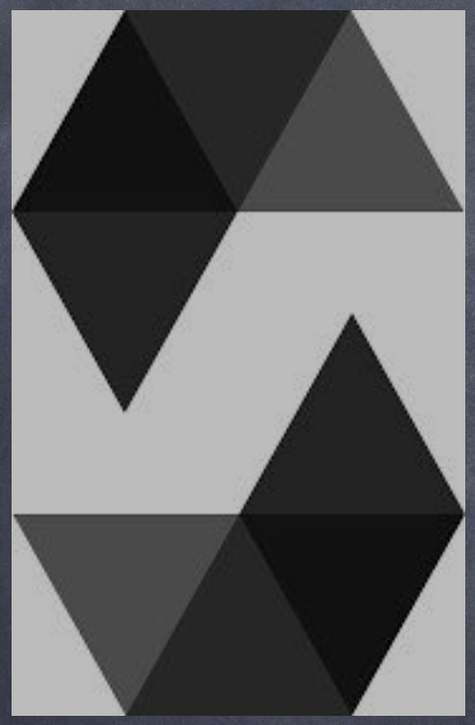
Added/Removed/Checked
-> O(1)

Enumerated -> O(n)

Supported Mapping Type
uint256 => address

#179

OZ
EnumerableSet



Managing Sets

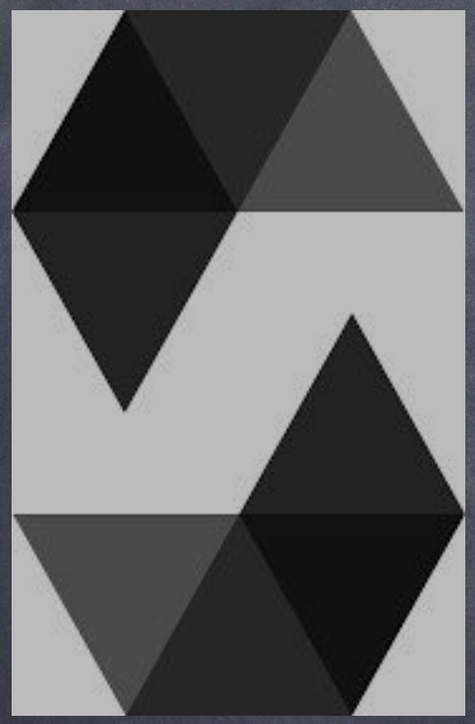
Added/Removed/Checked
-> O(1)

Enumerated -> O(n)

Supported Set Types
bytes/address/uint256

#180

OZ
BitMaps



Managing BitMaps
uint256 -> bool

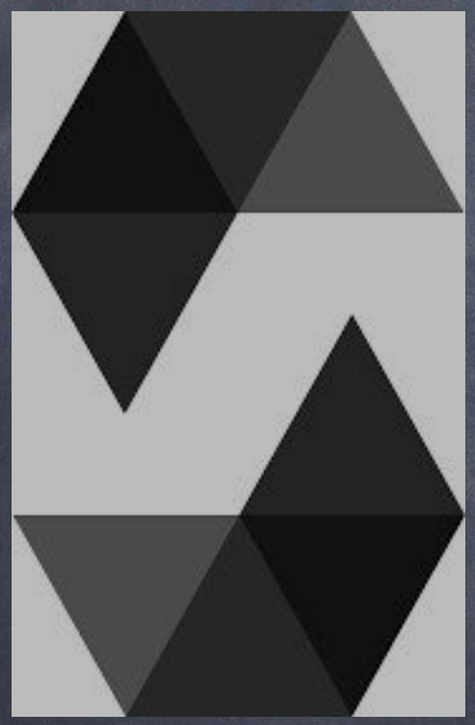
get(bitmap, index) → bool

setTo(bitmap, index, value)
set (bitmap, index)

unset(bitmap, index)

#181

OZ
PaymentSplitter



Split Ether Payments
Group of Accounts

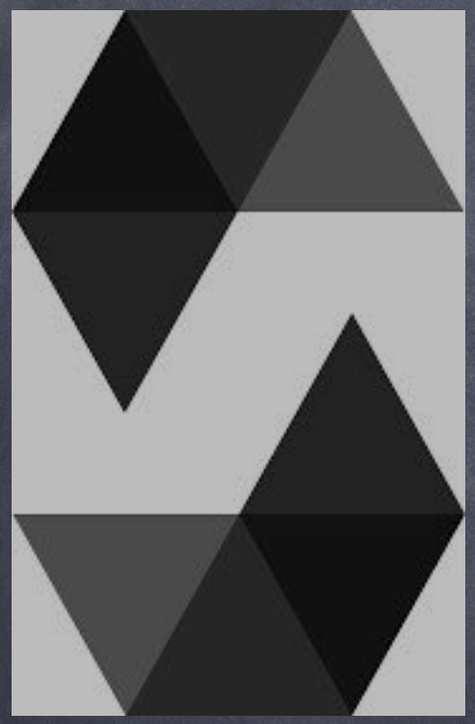
Sender Agnostic
Split -> Equal/Arbitrary

Account -> Shares
Claim -> Proportional

Pull Payment Model

#182

OZ
TimelockController



Enforces Timelock
onlyOwner Operations

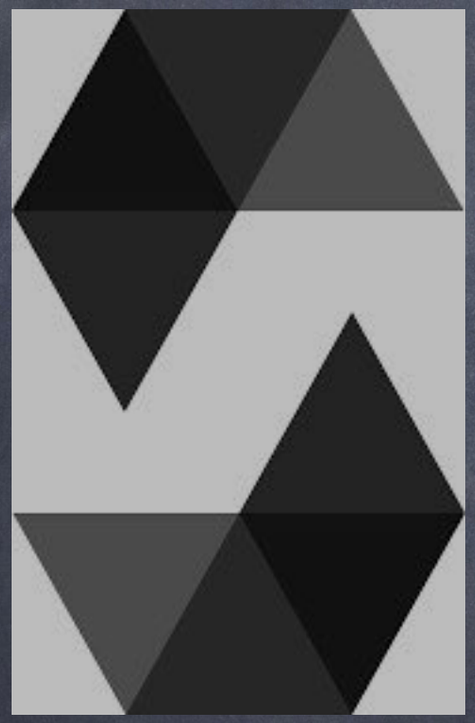
Users -> Exit
Operations -> Apply

Operations -> Schedule/
Delay/Execute/Cancel/Batch

Operations -> Pending/
Ready/Done/updateDelay

#183

OZ
ERC2771Context



ERC-2771
Meta-transactions

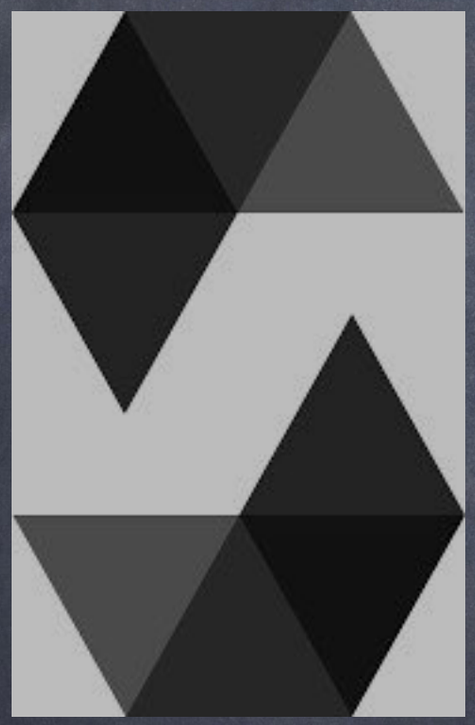
Authorized -> Tx Signer
Relayer -> Gas Pay

Tx Signer -> Gas Relay
Gas Relay -> Trusted FWDer

Trusted FWDer -> Contract
Contract -> ERC-2771

#184

OZ
MinimalForwarder



ERC-2771
Meta-transactions

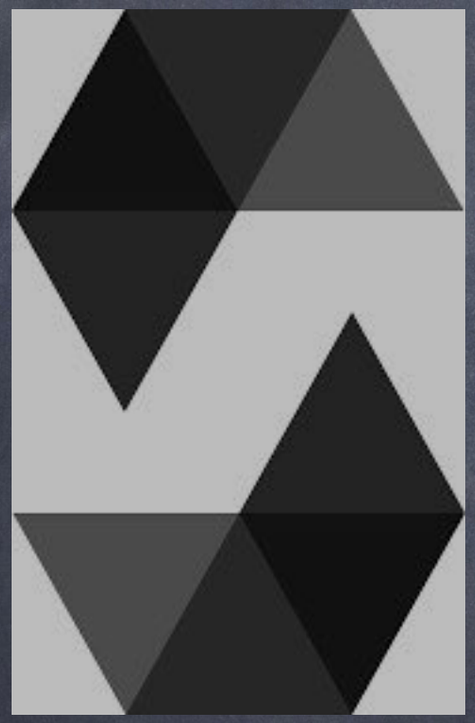
Simple Minimal Forwarder

Nonce & Signature Checks

verify (req, signature)
execute (req, signature)

#185

OZ
Proxy



Proxy -> Implementation
delegateCall

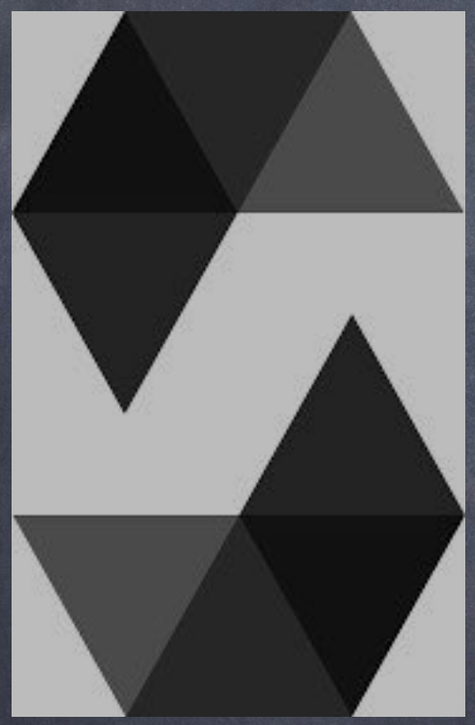
_fallback() ->
_implementation()

_delegate(address
implementation)

_beforeFallback()
Hook -> Delegate

#186

OZ
ERC1967Proxy



Upgradeable Proxy
Implementation -> Changed

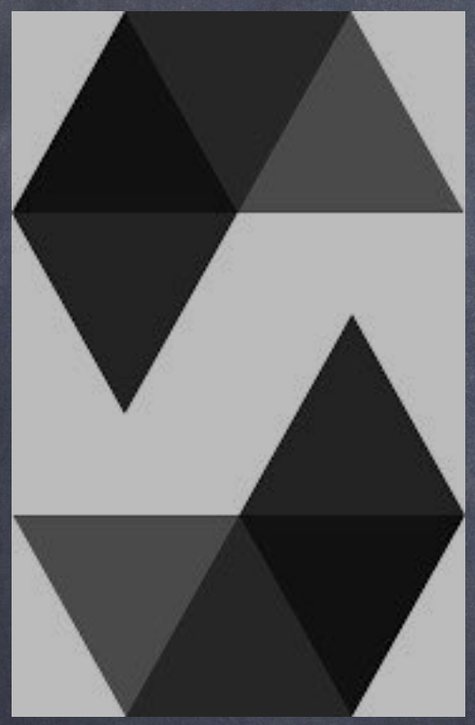
Data(Proxy) -> Logic(Impl)
Storage Conflict

constructor(address _logic,
bytes _data)

_upgradeTo(address
newImplementation)

#187

OZ TransparentUpgradeableProxy



Upgradeable -> Admin
Selector Clash -> Attack

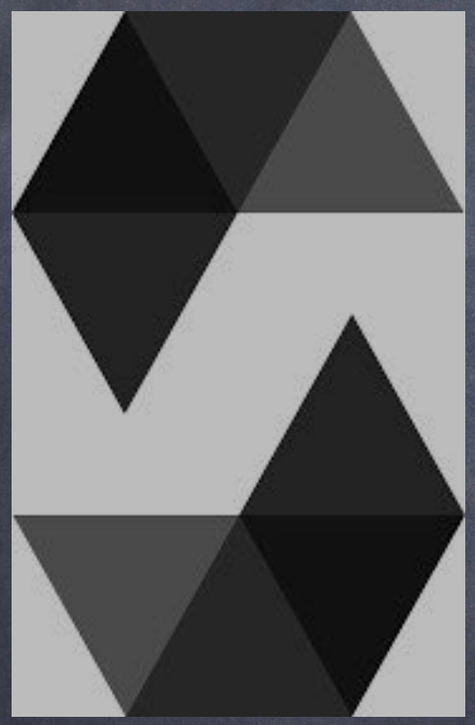
Non-Admin -> Implementation
Even Calls -> Match Proxy

Admin -> Proxy Admin Fns
Calls X-> Implementation

Admin -> Admin Actions
Upgrade -> Impl or Admin

#188

OZ ProxyAdmin



Admin Contract -> TransparentUpgradeableProxy

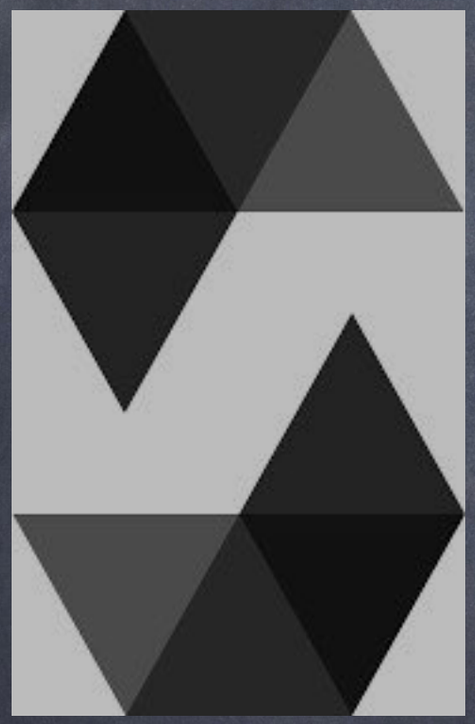
getProxyImplementation()
getProxyAdmin()

changeProxyAdmin()
upgrade(proxy, implementation)

upgradeAndCall(proxy, implementation, data)

#189

OZ
BeaconProxy



Implementation Addr ->
UpgradeableBeacon

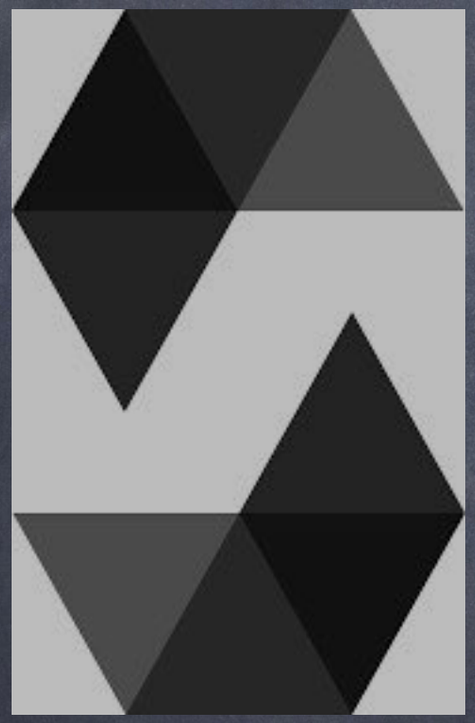
Beacon Address -> Slot
uint256(keccak256('eip1967.proxy.
beacon')) - 1

Constructor -> Beacon Init
_beacon() -> Beacon Addr

_implementation()
_setBeacon(beacon, data)

#190

OZ
UpgradeableBeacon



UpgradeableBeacon ->
BeaconProxy

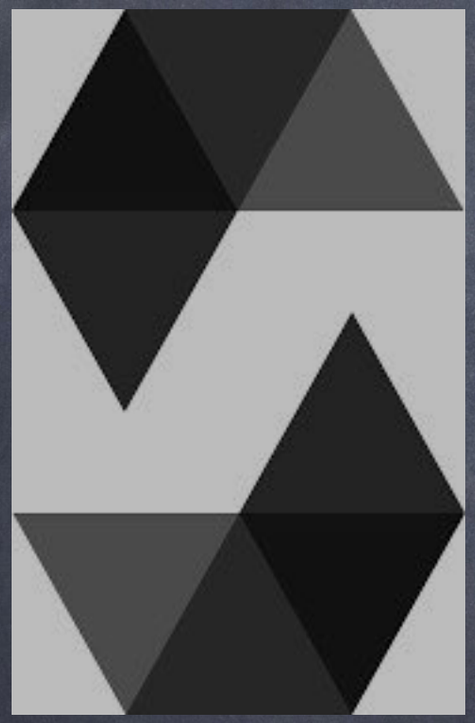
Owner -> Change Impl

Constructor -> Impl Init
Deployer -> Owner

_implementation()
_upgradeTo(newImplementation)

#191

OZ
Clones



EIP-1167
Minimal Proxy Contracts

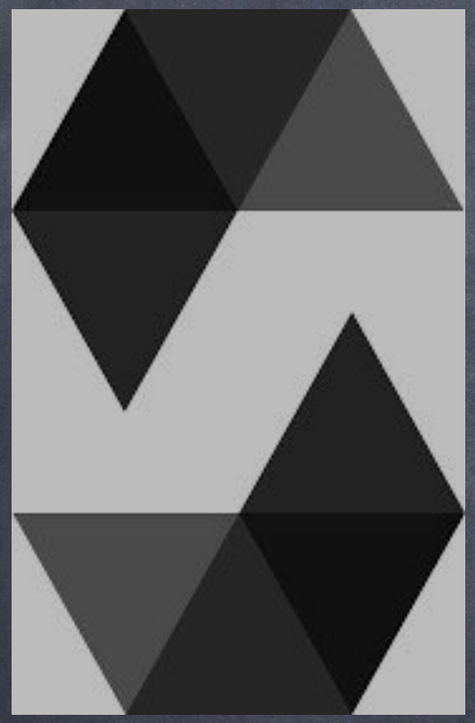
Delegate -> Fixed Impl
Deploy -> Create/Create2

clone(implementation) → instance

cloneDeterministic(implementation, salt) → instance

#192

OZ
Initializable



Proxy -> Impl
Data -> Logic

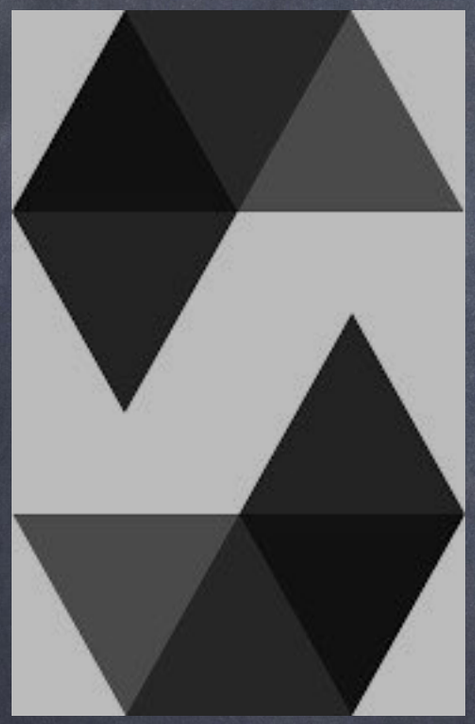
Constructor -> Initialize
All Contracts -> Inheritance

Once & Immediate

initializer Modifier
Immediate -> Deployment

#193

Dappsys
DSProxy



Proxy -> Impl

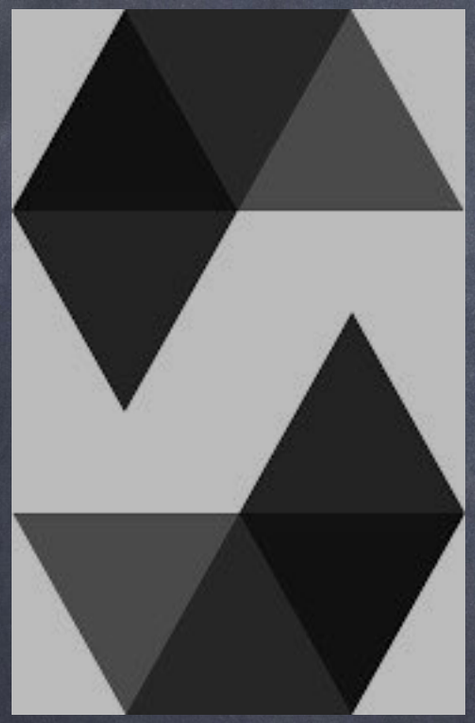
Contract Bytecode
Function Calldata

Create + Delegatecall

DSProxyFactory &
DSProxyCache

#194

Dappsys
DSMath



Arithmetic Functions X->
Overflow/Underflow

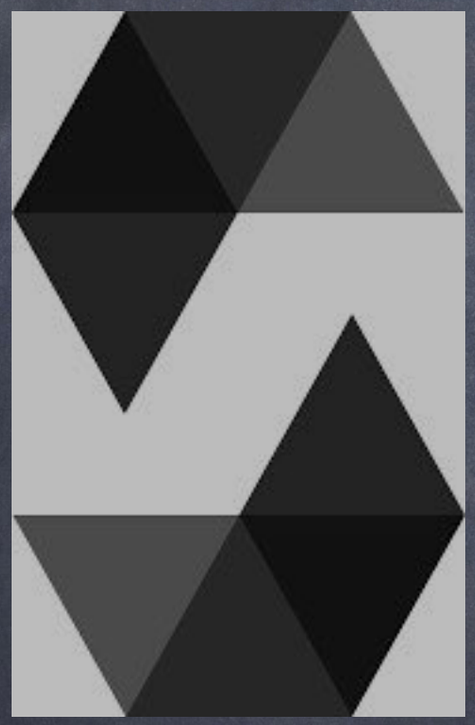
add, sub, mul, min, max
Fixed-point Math

Wad -> 18 Decimals
Ray -> 27 Decimals

Wad -> wmul, wdiv
Ray -> rmul, rdiv, rpow

#195

Dappsys
DSAuth



Authorization

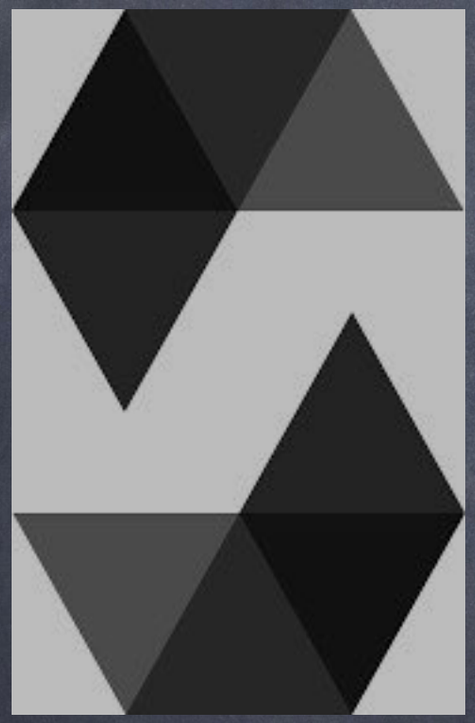
auth Modifier
isAuthorized()

msg.sender -> Owner
msg.sender -> Contract

msg.sender -> authority

#196

Dappsys
DSGuard



ACL
srcAddr, dstAddr -> fnSig

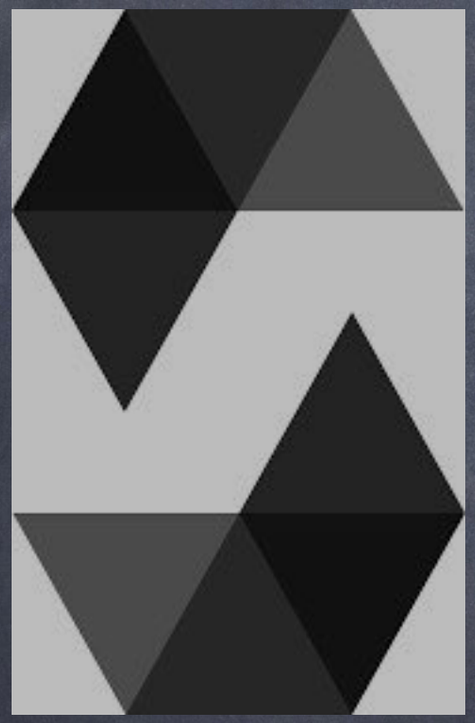
DSAuth authority
canCall() -> bool

[src][dst][sig] => boolean

src -> msg.sender, dst ->
Contract, sig -> Fn

#197

Dappsys
DSRoles



RBAC
ACLs -> Roles & Capabilities

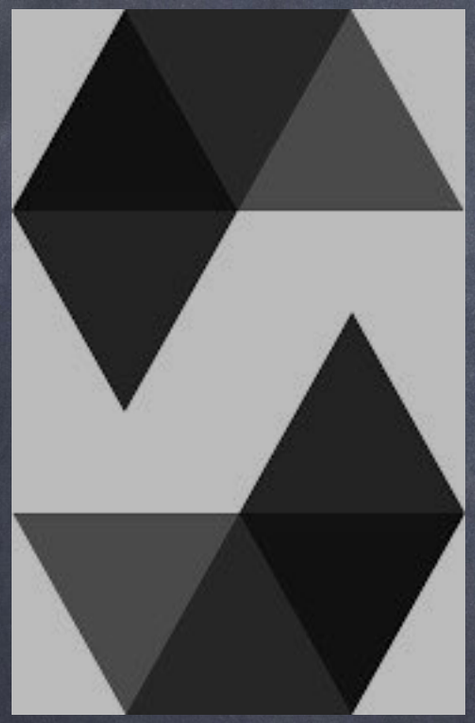
canCall() -> bool
User -> Fn @ Addr

Root Users
Public Capabilities

Role Capabilities -> Not
Root & Not Public

#198

WETH



Wrapper Ether
Ether Vs ERC-20

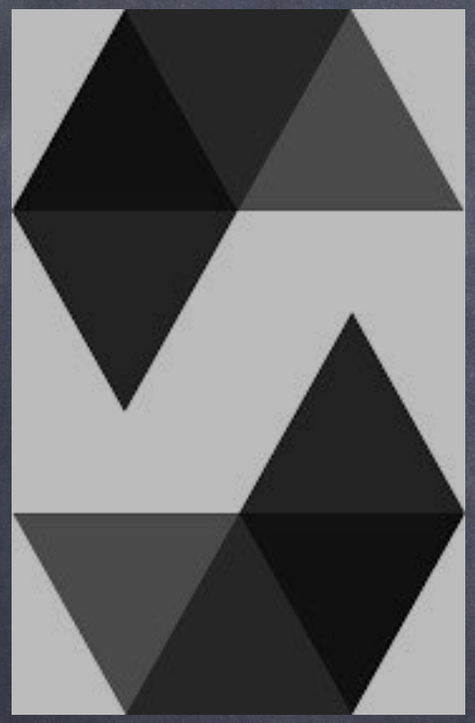
Wrapping -> ERC-20
Unwrapping -> Ether

WETH Contract -> 1:1
WETH9 -> 7M Ether

WETH10 -> Efficiency &
EIP-3156 Flash Loans

#199

Uniswap V2



AMM
x*y=k (x,y=token balances)

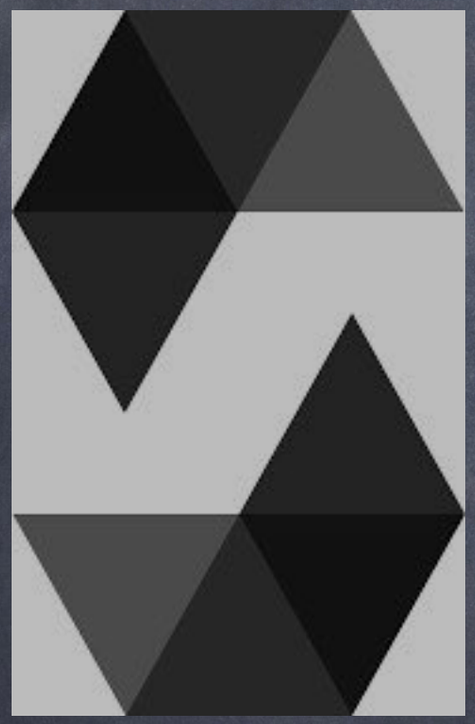
Pools, Token Pair
LP & LP Tokens

Swaps -> Token Pair
Maintain x*y=k

Oracles -> TWAPs

#200

Uniswap V3



Concentrated Liquidity
Custom Price Ranges

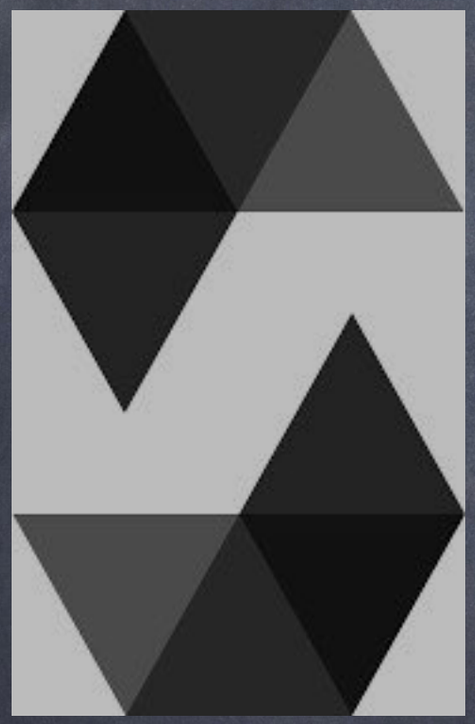
Capital Efficiency

Flexible Fees
0.05%, 0.30%, and 1.00%

Advanced TWAP
Cumulative Sums: One -> Array

#201

Chainlink



Oracles & Price Feeds

Offchain Data Providers -> Onchain Feeds

Oracles -> Chainlink N/W Aggregated Data

AggregatorV3Interface