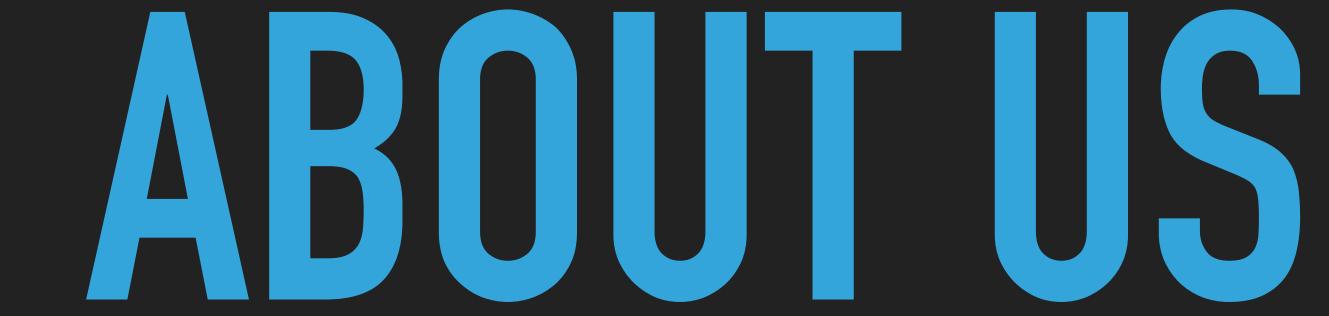
THINK LIKE AN ATTACKER BRINGING SMART CONTRACTS TO THEIR BREAK(ING) POINT











ABOUT US

Ilias

Co-Founder of Neodyme

Twitter: @A2nkF_



Thomas

- Co-Founder of Neodyme
- Twitter: @t_lambertz

ABOUT THIS WORKSHOP





ABOUT THIS WORKSHOP FERNKING EAN



ABOUT THIS WORKSHOP THNKNG' LKEAN (LIST OF COMMON BUGS ON BLOG.NEODYME.IO

IS THIS BROKEN?



STHS BROKEN?



STHS BROKEN?

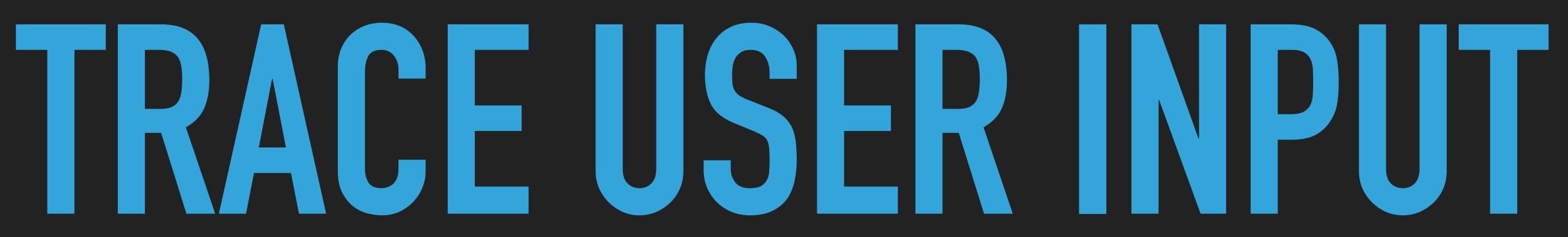
HOW DO I BREAK THIS?















FOR ATTACKERS



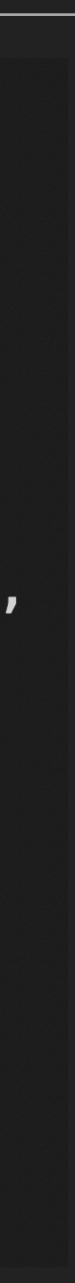






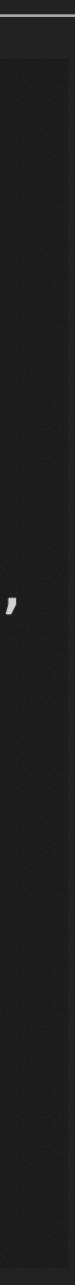
Nd

```
pub struct AccountInfo<'a> {
   pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
    pub lamports: Rc<RefCell<&'a mut u64>>,
    pub data: Rc<RefCell<&'a mut [u8]>>,
    pub owner: &'a Pubkey,
    pub executable: bool,
   // [...]
```





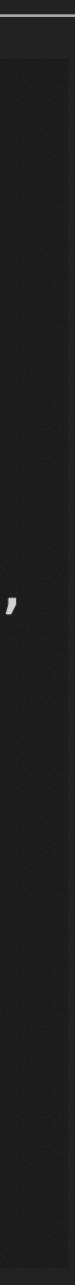
```
pub struct AccountInfo<'a> {
 pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
   pub lamports: Rc<RefCell<&'a mut u64>>,
   pub data: Rc<RefCell<&'a mut [u8]>>,
   pub owner: &'a Pubkey,
   pub executable: bool,
   // [...]
```



Used to identify an account

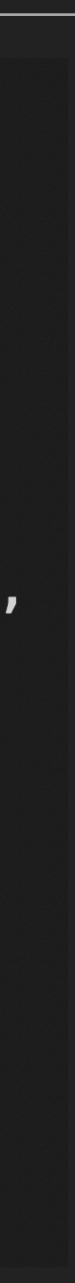


```
pub struct AccountInfo<'a> {
pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
   pub lamports: Rc<RefCell<&'a mut u64>>,
   pub data: Rc<RefCell<&'a mut [u8]>>,
   pub owner: &'a Pubkey,
   pub executable: bool,
   // [...]
```





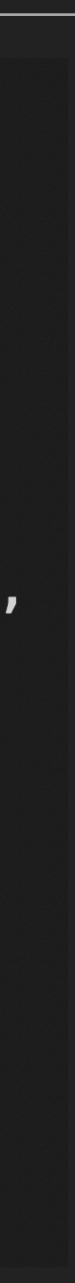
<pre>struct AccountInfo<'a> { pub key: &'a Pubkey,</pre>
<pre>pub is_signer: bool,</pre>
// []
<pre>pub lamports: Rc<refcell<&'a mut="" u64="">></refcell<&'a></pre>
<pre>pub data: Rc<refcell<&'a [u8]="" mut="">>,</refcell<&'a></pre>
pub owner: &'a Pubkey,
<pre>pub executable: bool,</pre>
// []



Flag indicating if the account has signed the transaction

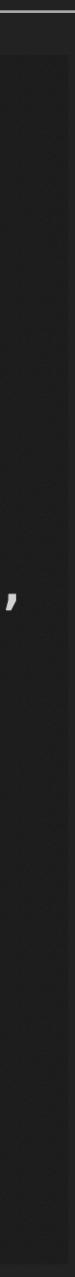


<pre>struct AccountInfo<'a> { pub key: &'a Pubkey,</pre>
<pre>pub is_signer: bool,</pre>
// []
<pre>pub lamports: Rc<refcell<&'a mut="" u64="">></refcell<&'a></pre>
<pre>pub data: Rc<refcell<&'a [u8]="" mut="">>,</refcell<&'a></pre>
pub owner: &'a Pubkey,
<pre>pub executable: bool,</pre>
// []



Nd

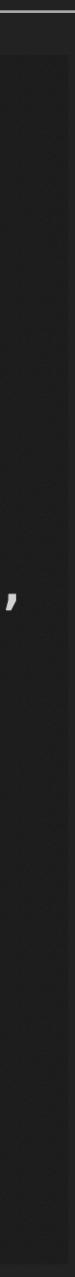
```
pub struct AccountInfo<'a> {
   pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
   pub lamports: Rc<RefCell<&'a mut u64>>,
    pub data: Rc<RefCell<&'a mut [u8]>>,
    pub owner: &'a Pubkey,
    pub executable: bool,
   // [...]
```



- Account balance in lamports
- 1 SOL = 1_000_000_000
 lamports
- Runtime asserts total_lamports_before == total_lamports_after

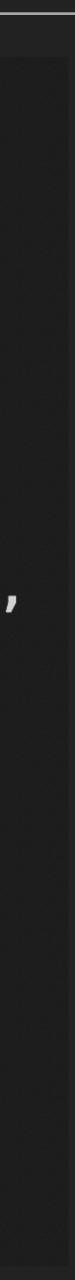


```
pub struct AccountInfo<'a> {
   pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
   pub lamports: Rc<RefCell<&'a mut u64>>,
   pub data: Rc<RefCell<&'a mut [u8]>>,
   pub owner: &'a Pubkey,
   pub executable: bool,
   // [...]
```



Nd

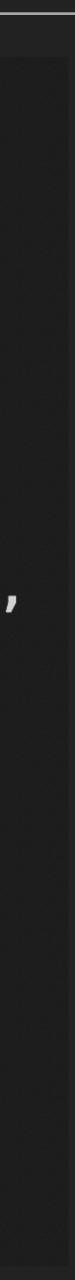
```
pub struct AccountInfo<'a> {
    pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
    pub lamports: Rc<RefCell<&'a mut u64>>,
   pub data: Rc<RefCell<&'a mut [u8]>>,
    pub owner: &'a Pubkey,
    pub executable: bool,
   // [...]
```



- Up to 10MB of mutable storage
- Can only be written by the 'owner' account
- Can not be resized (currently)

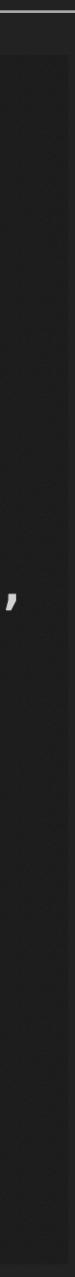


```
pub struct AccountInfo<'a> {
   pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
   pub lamports: Rc<RefCell<&'a mut u64>>,
   pub data: Rc<RefCell<&'a mut [u8]>>,
   pub owner: &'a Pubkey,
   pub executable: bool,
   // [...]
```



Nd

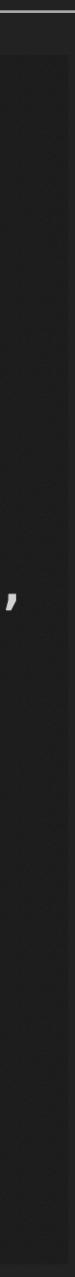
```
pub struct AccountInfo<'a> {
   pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
    pub lamports: Rc<RefCell<&'a mut u64>>,
    pub data: Rc<RefCell<&'a mut [u8]>>,
   pub owner: &'a Pubkey,
    pub executable: bool,
   // [...]
```



- The only account that can write to the data
- It can only be changed if the data is zero
- ▶ !!! CHECK THE OWNER !!!



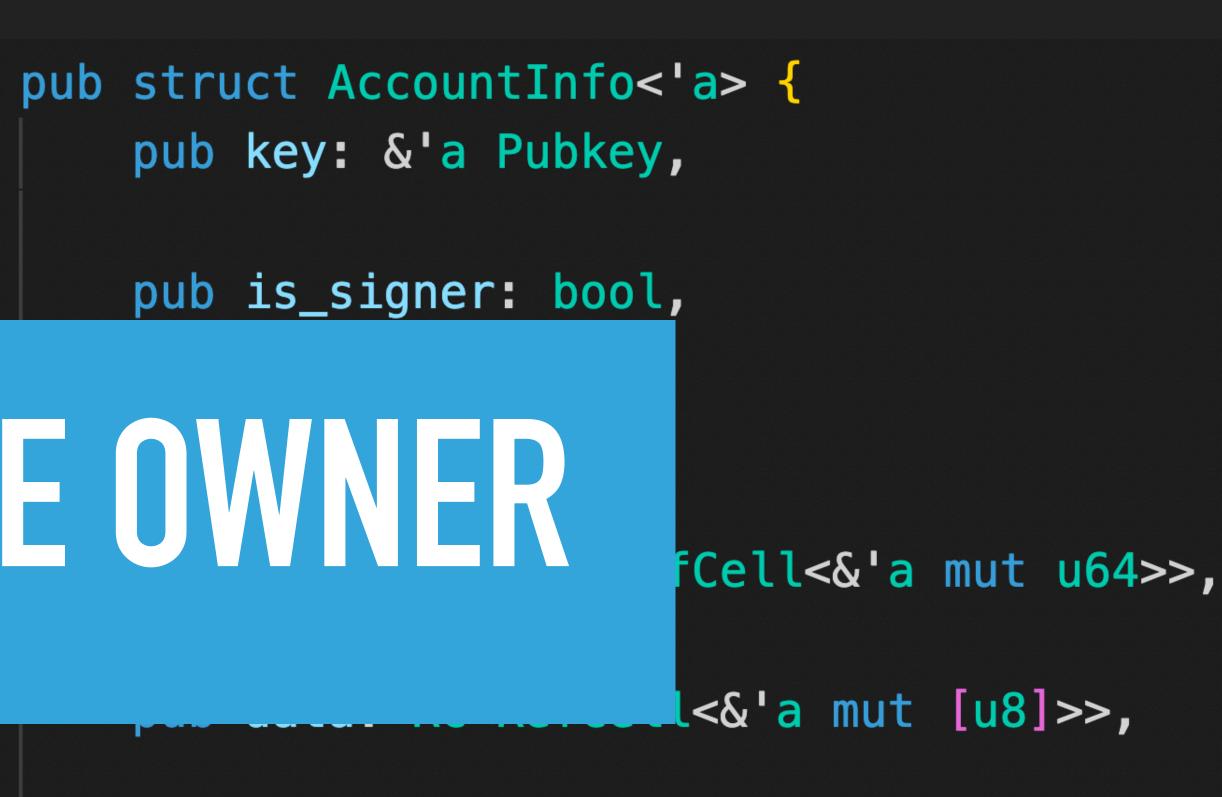
```
pub struct AccountInfo<'a> {
   pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
   pub lamports: Rc<RefCell<&'a mut u64>>,
   pub data: Rc<RefCell<&'a mut [u8]>>,
   pub owner: &'a Pubkey,
   pub executable: bool,
   // [...]
```



- The only account that can write to the data
- It can only is zero
- ► !!! CHECK

CHECK THE OWNER

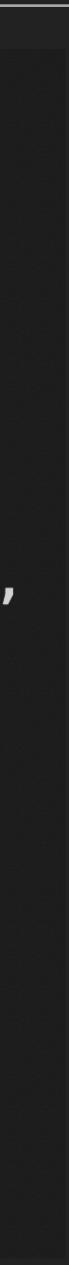




pub owner: &'a Pubkey,

pub executable: bool,

// [...]

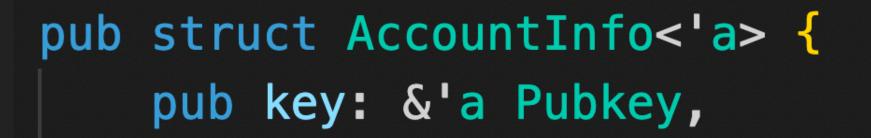


an write

It ca. is zero

► !!! CHECK





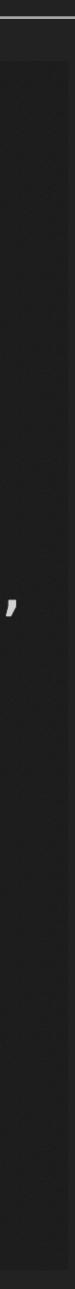
pub is_signer: bool,

FCell<&'a mut u64>>,

<&'a mut [u8]>>,

Jkey,

bool,



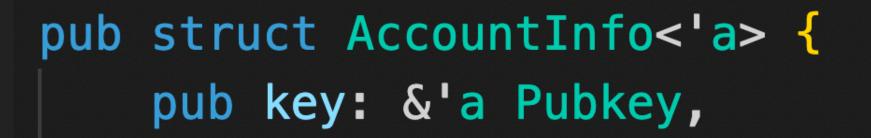
The c to t'

an write



CHECK





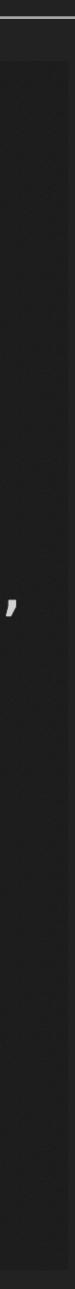
pub is_signer: bool,

fCell<&'a mut u64>>,

l<&'a mut [u8]>>,

Jkey,

bool,



The c to t'

an write



CHECK





The c to to to

an write



CHECK



pub struct AccountInfo<'a> {
 pub key: &'a Pubkey,

pub is_signer: bool,

THE

OWNER

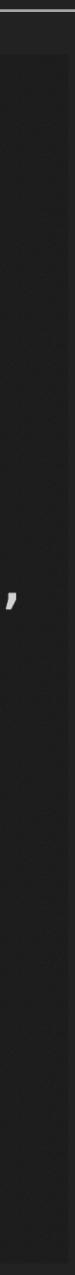
bool,



<u>3</u>]>>,

Nd

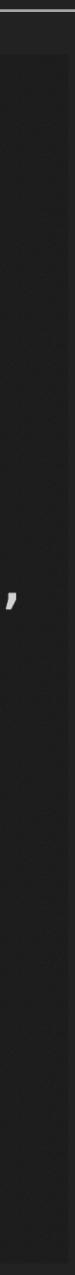
```
pub struct AccountInfo<'a> {
    pub key: &'a Pubkey,
   pub is_signer: bool,
   // [...]
    pub lamports: Rc<RefCell<&'a mut u64>>,
    pub data: Rc<RefCell<&'a mut [u8]>>,
    pub owner: &'a Pubkey,
    pub executable: bool,
   // [...]
```



Indicates if this account is a smart contract or not



```
pub struct AccountInfo<'a> {
    pub key: &'a Pubkey,
    pub is_signer: bool,
   // [...]
    pub lamports: Rc<RefCell<&'a mut u64>>,
    pub data: Rc<RefCell<&'a mut [u8]>>,
    pub owner: &'a Pubkey,
    pub executable: bool,
    // [...]
```











Recent Blockhash: BKWbZbqjVTuAHwZO Signature 0: 21vhXxS1rRzhrZAs Signature 1: 4g5A8oMozkFW3sSw
Account 0: srw- BWd4RBYcuk3edUePUy
Account 1: srw- KoooVyhdpoRPA6gpn7 Account 2: -rw- B6Zm6Jy1c9iEMCURbh
Account 3: -rw- 8Br27saLgmjGAj2uHE Account 4: -r 111111111111111111
Account 5: -r-x W4113t33333333333333
Instruction 0 Program: W4113t333333333333333
Account 0: B6Zm6Jy1c9iEMCURbho Account 1: 8Br27saLgmjGAj2uHBk
Account 2: KoooVyhdpoRPA6gpn7> Account 3: B6Zm6Jy1c9iEMCURbhc
Account 4: 111111111111111111111
Data: [2, 0, 128, 198, 164, 12 Status: Ok

GiVFCDREyiMAYGuufEqhWDHcrPEgp

Recent Blockhash: BKWbZbqjVTuAHwZGiVFCDREyiMAYGuufEqhWDHcrPEgp Signature 0: 21vhXxS1...rRzhrZAs Signature 1: 4g5A8oMo...zkFW3sSw Account 0: srw- BWd4RBYcuk3edUePUy9kKpSv8WuobQojEUCDasTjPgne (fee payer) Account 1: srw- KoooVyhdpoRPA6gpn7xr3cmjqAvtpHcjcBX6JBKu1nf Account 2: -rw- B6Zm6Jy1c9iEMCURbhdA3Quqd5PGoWjBe8ZX57TsLaj7 Account 3: -rw- 8Br27saLgmjGAj2uHBK4tXgivsrHFT2jQqAvypqb8USm Instruction 0 Program: Account 0: B6Zm6Jy1c9iEMCURbhdA3Quqd5PGoWjBe8ZX57TsLaj7 (2) Account 1: 8Br27saLgmjGAj2uHBK4tXgivsrHFT2jQqAvypqb8USm (3) Account 2: KoooVyhdpoRPA6gpn7xr3cmjqAvtpHcjcBX6JBKu1nf (1) Account 3: B6Zm6Jy1c9iEMCURbhdA3Quqd5PGoWjBe8ZX57TsLaj7 (2) Data: [2, 0, 128, 198, 164, 126, 141, 3, 0] Status: Ok

Recent Blockhash: BKWbZbqjVTuAHwZGiVFCDREyiMAYGuufEqhWDHcrPEgp Signature 0: 21vhXxS1...rRzhrZAs Signature 1: 4g5A8oMo...zkFW3sSw Account 0: srw- BWd4RBYcuk3edUePUy9kKpSv8WuobQojEUCDasTjPgne (fee payer) Account 1: srw- KoooVyhdpoRPA6gpn7xr3cmjqAvtpHcjcBX6JBKu1nf Account 2: -rw- B6Zm6Jy1c9iEMCURbhdA3Quqd5PGoWjBe8ZX57TsLaj7 Account 3: -rw- 8Br27saLgmjGAj2uHBK4tXgivsrHFT2jQqAvypqb8USm Instruction 0 Program: Account 0: B6Zm6Jy1c9iEMCURbhdA3Quqd5PGoWjBe8ZX57TsLaj7 (2) Account 1: 8Br27saLgmjGAj2uHBK4tXgivsrHFT2jQqAvypqb8USm (3) Account 2: KoooVyhdpoRPA6gpn7xr3cmjqAvtpHcjcBX6JBKu1nf (1) Account 3: B6Zm6Jy1c9iEMCURbhdA3Quqd5PGoWjBe8ZX57TsLaj7 (2) Data: [2, 0, 128, 198, 164, 126, 141, 3, 0] Status: Ok

	Recent Blockhash: BKWbZbqjVTuAHwZ0 Signature 0: 21vhXxS1rRzhrZAs
	Signature 1: 4g5A8oMozkFW3sSw
	Account 0: srw- BWd4RBYcuk3edUePU
	Account 1: srw- KoooVyhdpoRPA6gpn7
	Account 2: -rw- B6Zm6Jy1c9iEMCURb
	Account 3: -rw- 8Br27saLgmjGAj2uH
	Account 4: -r 111111111111111111
	Account 5: -r-x W4113t333333333333333
	Instruction 0
	Program: W4113t3333333333333333333333333333333333
	Program: W4113t3333333333333333333333333333333333
	Program: W4113t33333333333333333 Account 0: B6Zm6Jy1c9iEMCURbho Account 1: 8Br27saLgmjGAj2uHBb
	Program: W4113t3333333333333333 Account 0: B6Zm6Jy1c9iEMCURbho Account 1: 8Br27saLgmjGAj2uHBb Account 2: KoooVyhdpoRPA6gpn7x
	Program: W4113t3333333333333333333333333333333333
1	Program: W4113t3333333333333333 Account 0: B6Zm6Jy1c9iEMCURbho Account 1: 8Br27saLgmjGAj2uHBb Account 2: KoooVyhdpoRPA6gpn7x

GiVFCDREyiMAYGuufEqhWDHcrPEgp

		: BKWbZbqjV hXxS1rRzł	
Signature	1: 4g5	A8oMozkFV	V3sSw
Account 0:	srw-	BWd4RBYcuk3e	edUePUy
Account 1:	srw-	KoooVyhdpoRF	PA6gpn7
Account 2:	- rw-	B6Zm6Jy1c9i	EMCURbł
Account 3:	- rw-	8Br27saLgmj(GAj2uHE
Account 4:	-r	1111111111111	
Account 5:	-r-x	W4113t333333	3333333
Instructio	n 0		
Progra	m: W	4113t3333333	3333333
Accoun	t 0: B	6Zm6Jy1c9iEN	1CURbhc
Accoun	t 1: 8	Br27saLgmjG/	Aj2uHBk
Accoun	t 2: K	oooVyhdpoRP/	۵6gpn7>
		6Zm6Jy1c9iEN	
		111111111111	
		128, 198, 1	164, 12
Status: Ok			

GiVFCDREyiMAYGuufEqhWDHcrPEgp

Recent Blockhash: BKWbZbgjVTuAHwZ
Signature 0: 21vhXxS1rRzhrZAs
Signature 1: 4g5A8oMozkFW3sSw
Account 0: srw- BWd4RBYcuk3edUePU
Account 1: srw- KoooVyhdpoRPA6gpn
Account 2: -rw- B6Zm6Jy1c9iEMCURb
Account 3: -rw- 8Br27saLgmjGAj2uH
Account 4: -r 11111111111111111
Account 5: -r-x W4113t333333333333333
Instruction 0
Program: W4113t3333333333333333333333333333333333
Account 0: B6Zm6Jy1c9iEMCURbh
Account 1, $QBr27cal ami(Ai)UHR$
Account 1: 8Br27saLgmjGAj2uHB
Account 2: KoooVyhdpoRPA6gpn7
Account 2: KoooVyhdpoRPA6gpn7
Account 2: KoooVyhdpoRPA6gpn7 Account 3: B6Zm6Jy1c9iEMCURbh

GiVFCDREyiMAYGuufEqhWDHcrPEgp

Recent Blockhash: BKWbZbgjVTuAHwZ0
Signature 0: 21vhXxS1rRzhrZAs
Signature 1: 4g5A8oMozkFW3sSw
Account 0: srw- BWd4RBYcuk3edUePUy
Account 1: srw- KoooVyhdpoRPA6gpn7
Account 2: -rw- B6Zm6Jy1c9iEMCURbh
Account 3: -rw- 8Br27saLgmjGAj2uHE
Account 4: -r 1111111111111111111
Account 5: -r-x W4113t33333333333333
Instruction 0
Program: W4113t3333333333333333
Account 0: B6Zm6Jy1c9iEMCURbho
Account 1: 8Br27saLgmjGAj2uHBk
Account 2: KoooVyhdpoRPA6gpn7>
Account 3: B6Zm6Jy1c9iEMCURbho
Account 4: 11111111111111111111
Data: [2, 0, 128, 198, 164, 12
Status: Ok

GiVFCDREyiMAYGuufEqhWDHcrPEgp









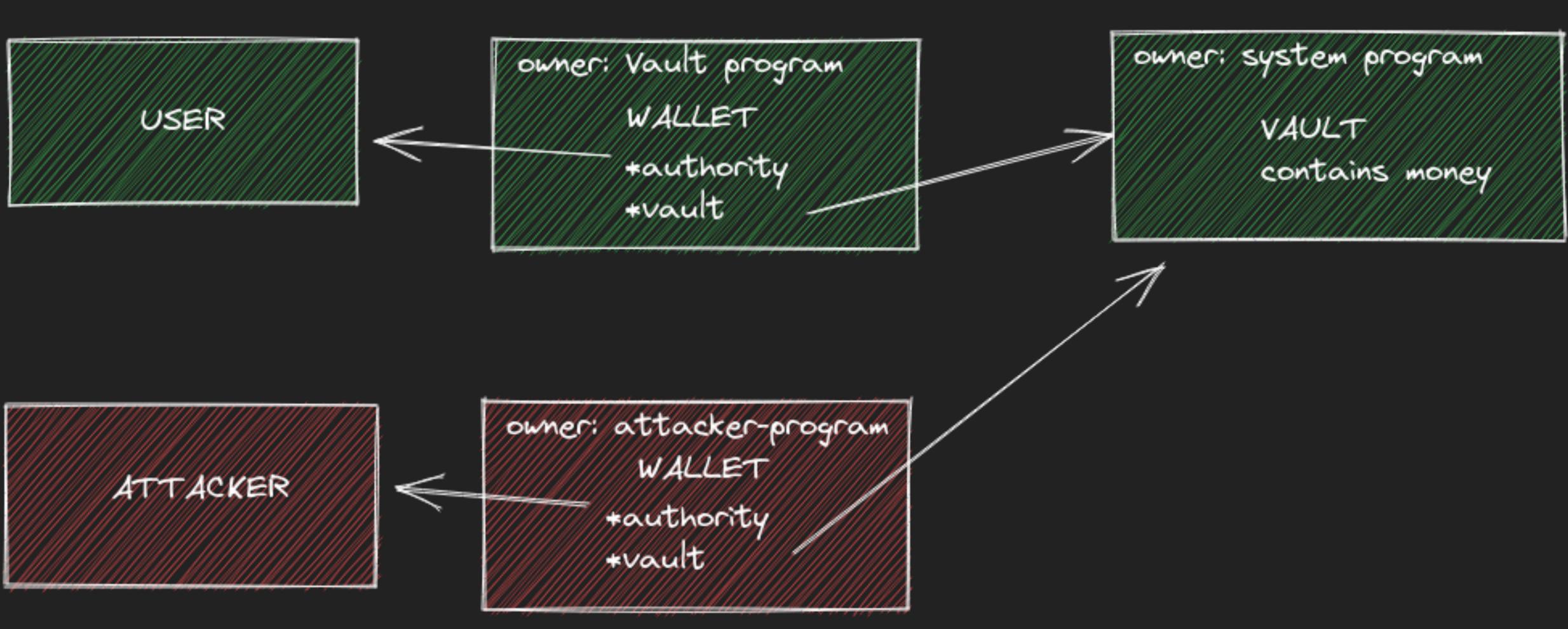


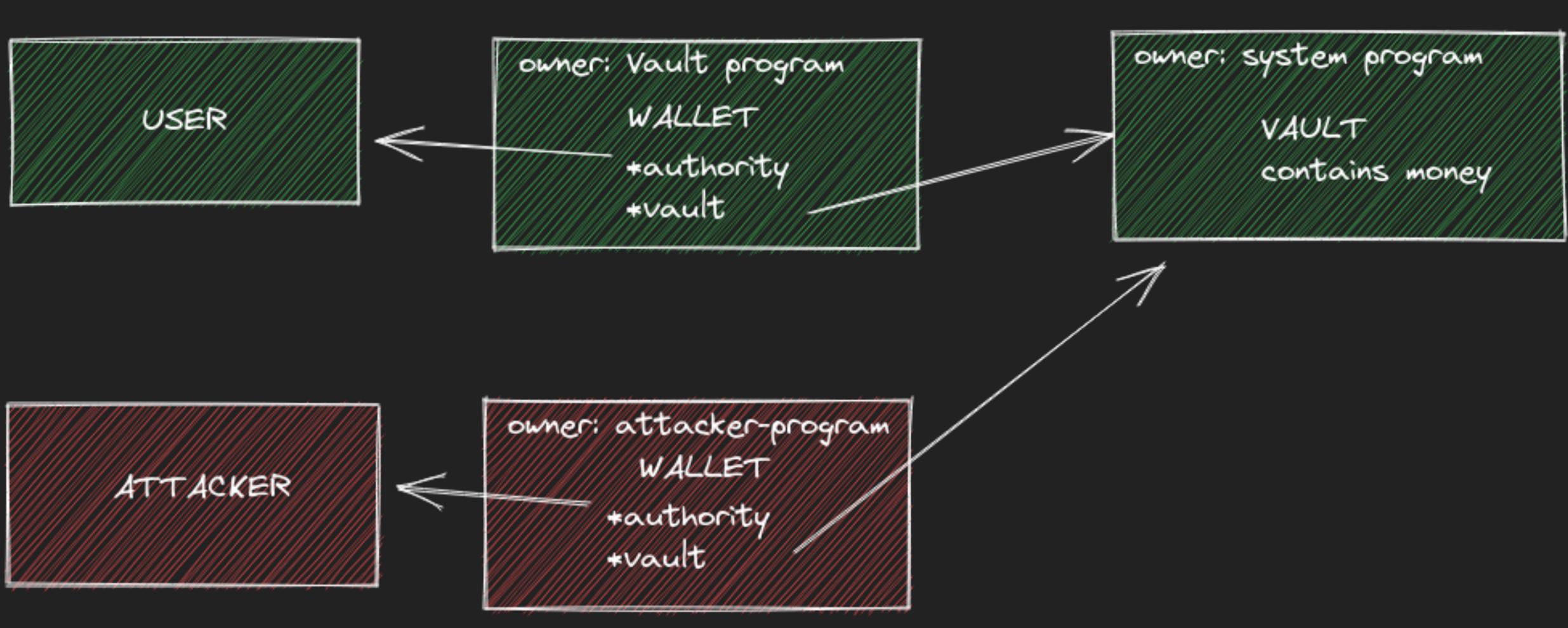
















Rust crate: <u>https://docs.rs/poc-framework/</u>



Visit: https://workshop.neodyme.io

We're providing remote environments but you can also use your own :)

