



Algorand Ranch PyTEAL - Workshop

The Power
of Change





Agenda

- **Setup**
- **Overview of Smart Contracts**
- **Walkthrough building a counter contract**
 - Step 1 - Setup Contract programs
 - Step 2 - Transaction type routing
 - Step 3 - Initialize the global variable
 - Step 4 - Handle input parameters
 - Step 5 - Implement add and deduct methods



Setup Sample

- Clone <https://github.com/algorand-devrel/workshop-pyteal>

```
cd workshop-pyteal
```

```
python -m venv .venv
```

```
source .venv/bin/activate
```

```
pip install git+https://github.com/algorand/pyteal
```

```
pip install algosdk
```

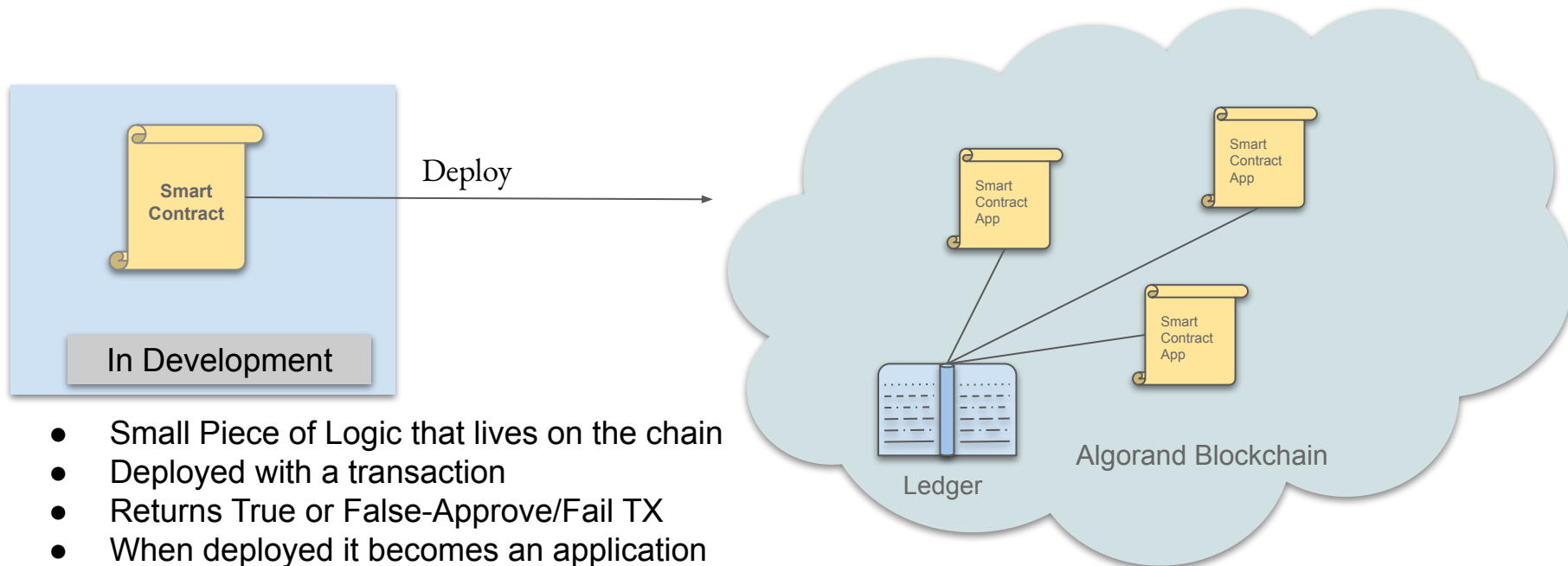
Make sure sandbox is up in a terminal `./sandbox up dev`

```
# Test Output
```

```
python3 -m src.complete.contract
```

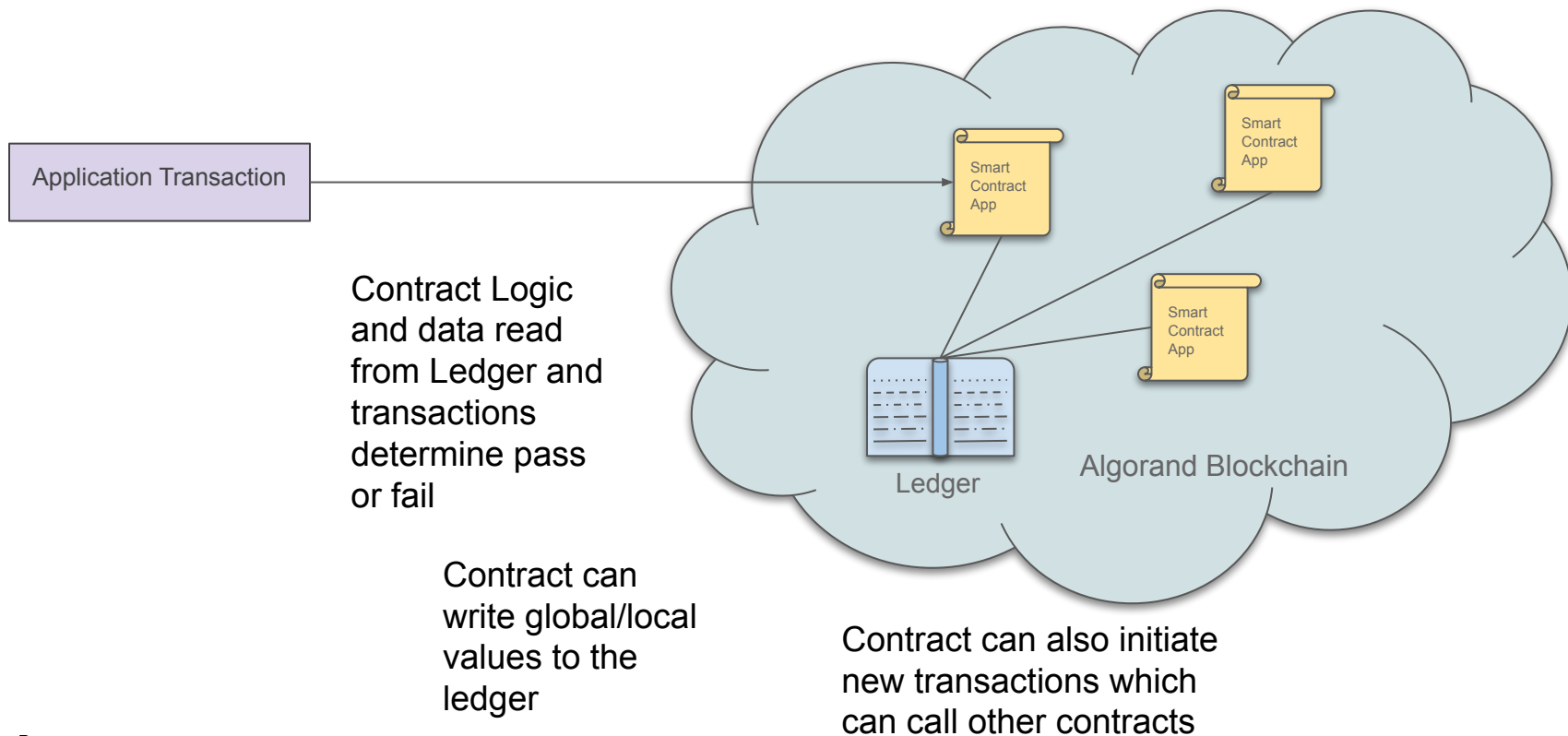


What is a Smart Contract



- Small Piece of Logic that lives on the chain
- Deployed with a transaction
- Returns True or False-Approve/Fail TX
- When deployed it becomes an application
 - Application ID and Address
- Written in TEAL, PyTEAL, or using the Reach Framework
- Executes on the AVM
- Supports Looping, Subroutines and Inner Transactions

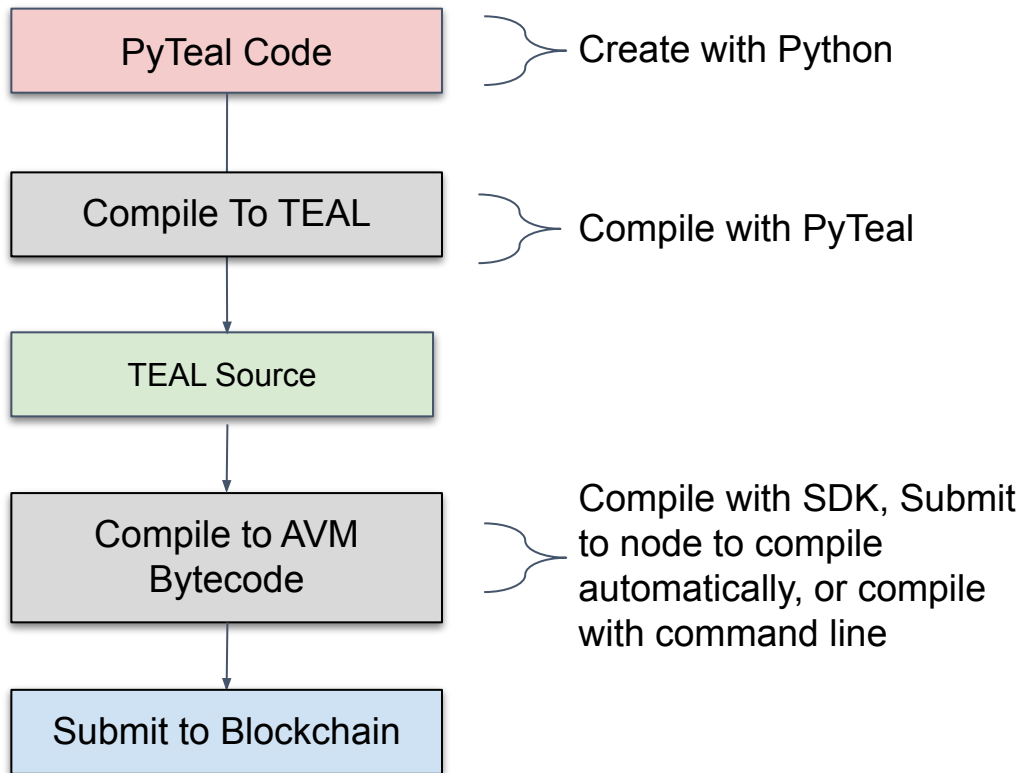
Smart Contract High Level





PyTeal

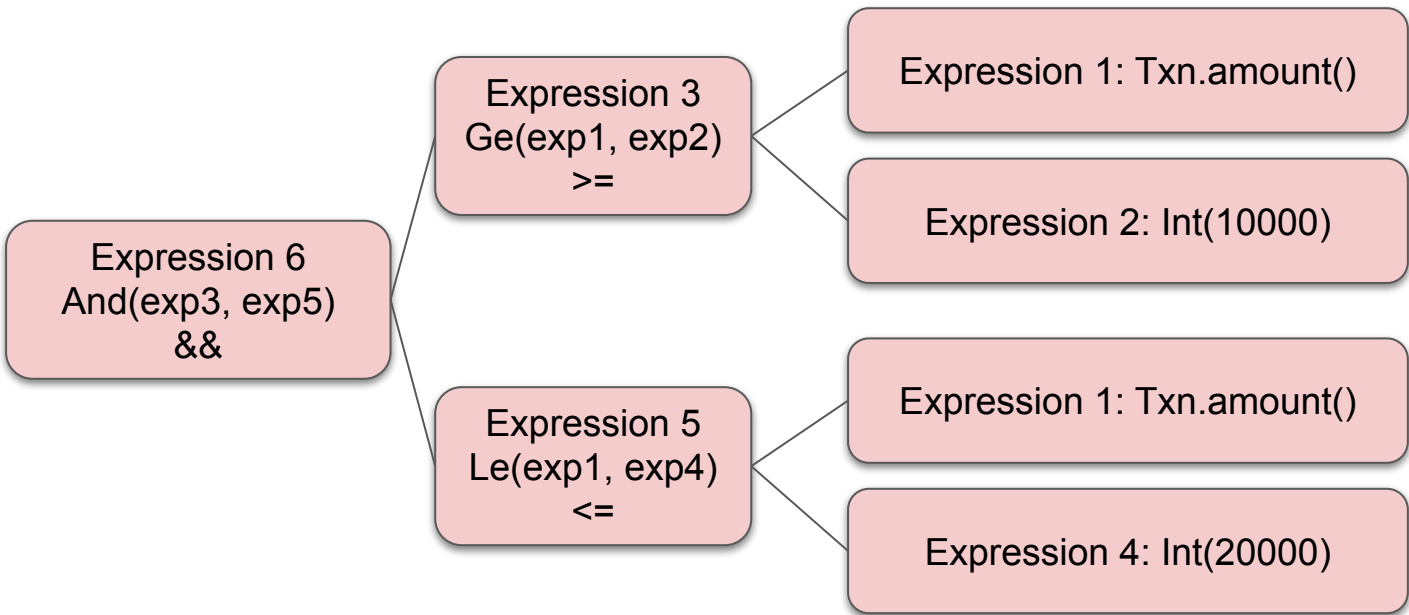
- **Python Library that produces TEAL**
- **TEAL compiled to bytecode and runs on Algorand AVM**





PyTeal Expression Tree Example

Transaction Amount ≥ 10000 and ≤ 20000



`class pyteal.TealType`

Bases: `enum.Enum`

Teal type enum.

`anytype= 2`

`bytes= 1`

`none= 3`

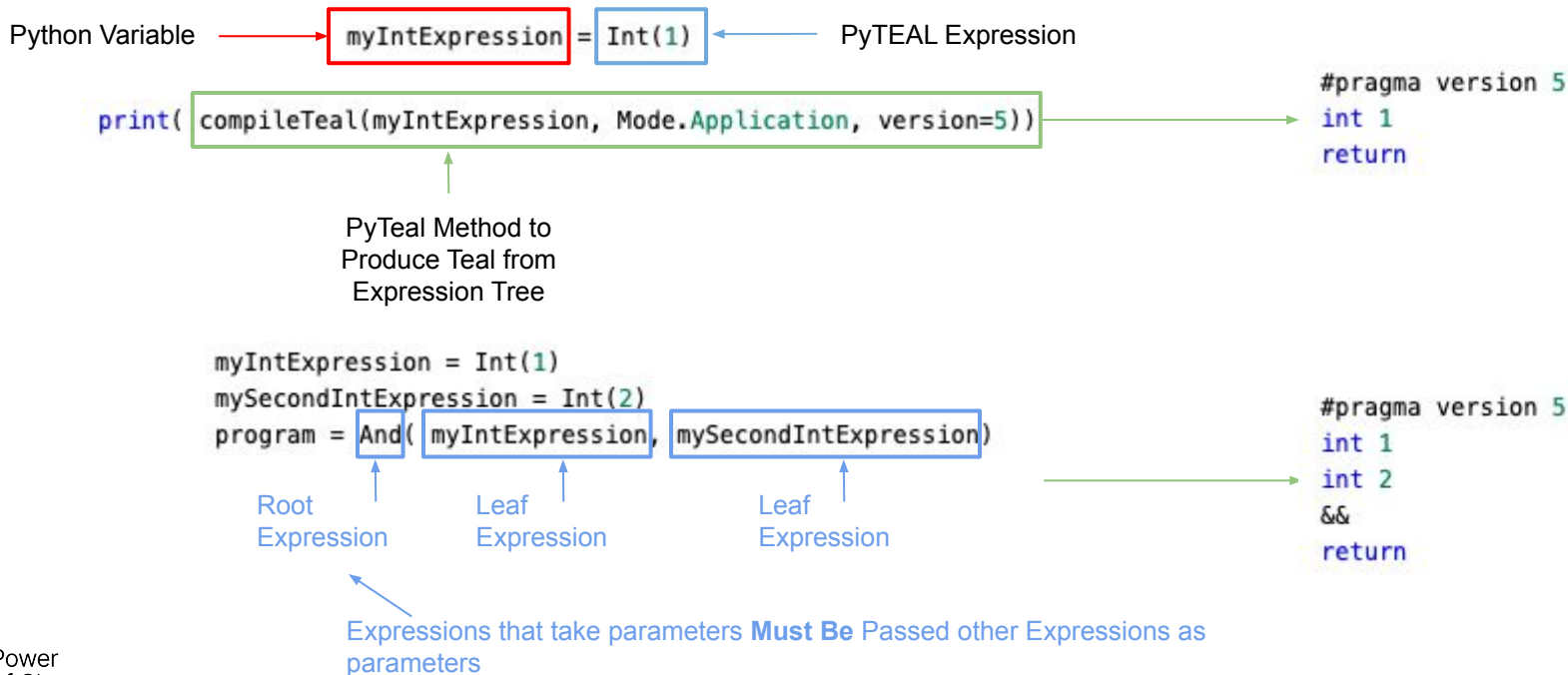
`uint64= 0`

`And(Txn.amount() >= Int(10000), Txn.amount() <= Int(20000))`



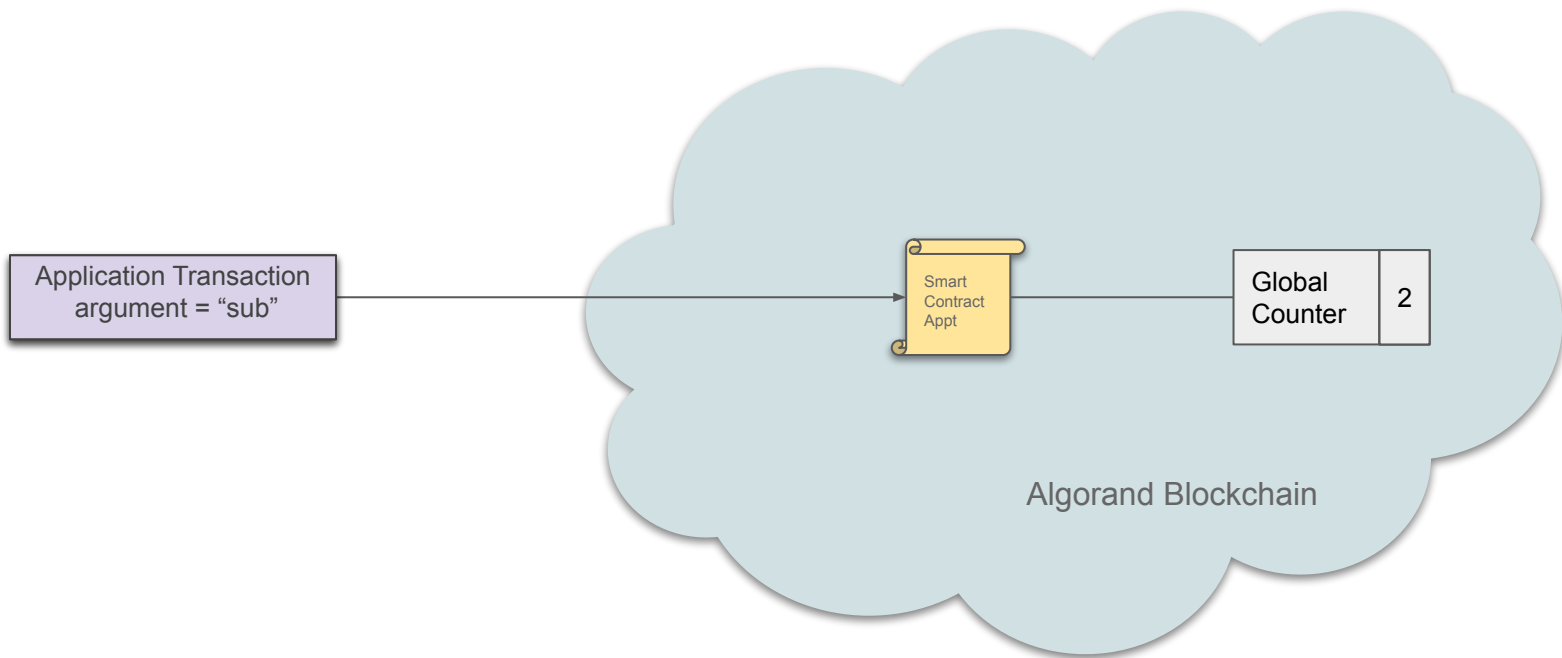
PyTeal Expressions

- Smart Contracts are built as an Expression Trees
- Expressions are created using PyTeal Class methods
- Class Methods take other Expressions as Parameters
- Expression are converted to TEAL when compiled



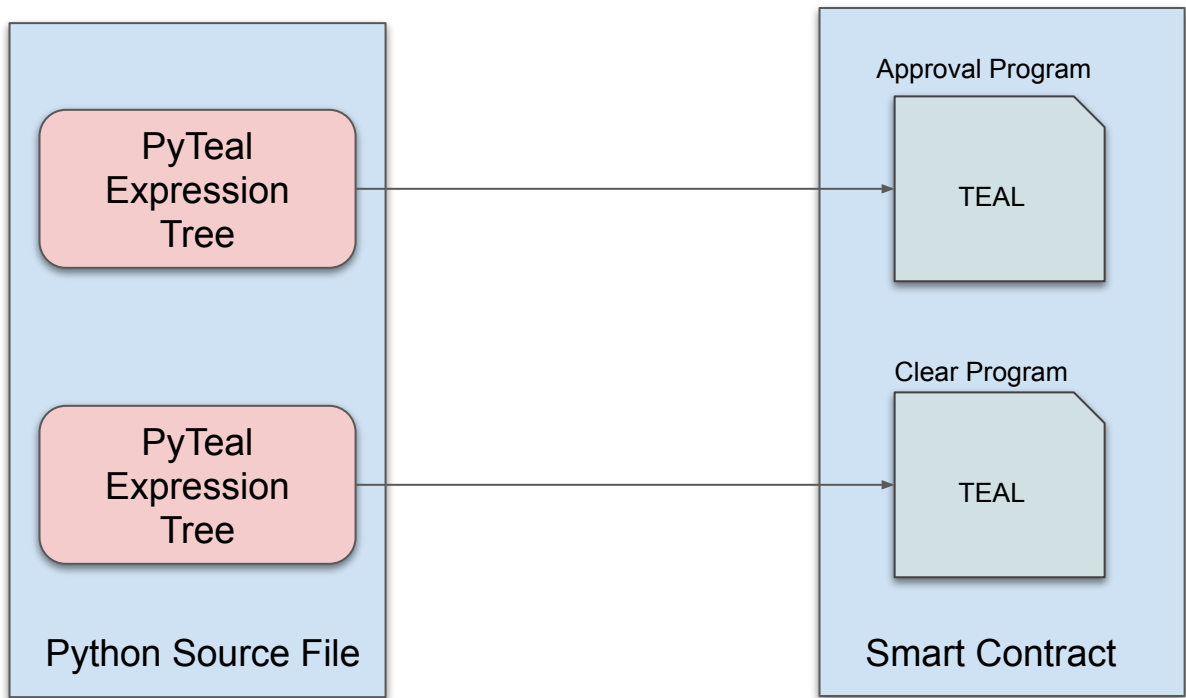


Incrementing Counter





Counter Step 1



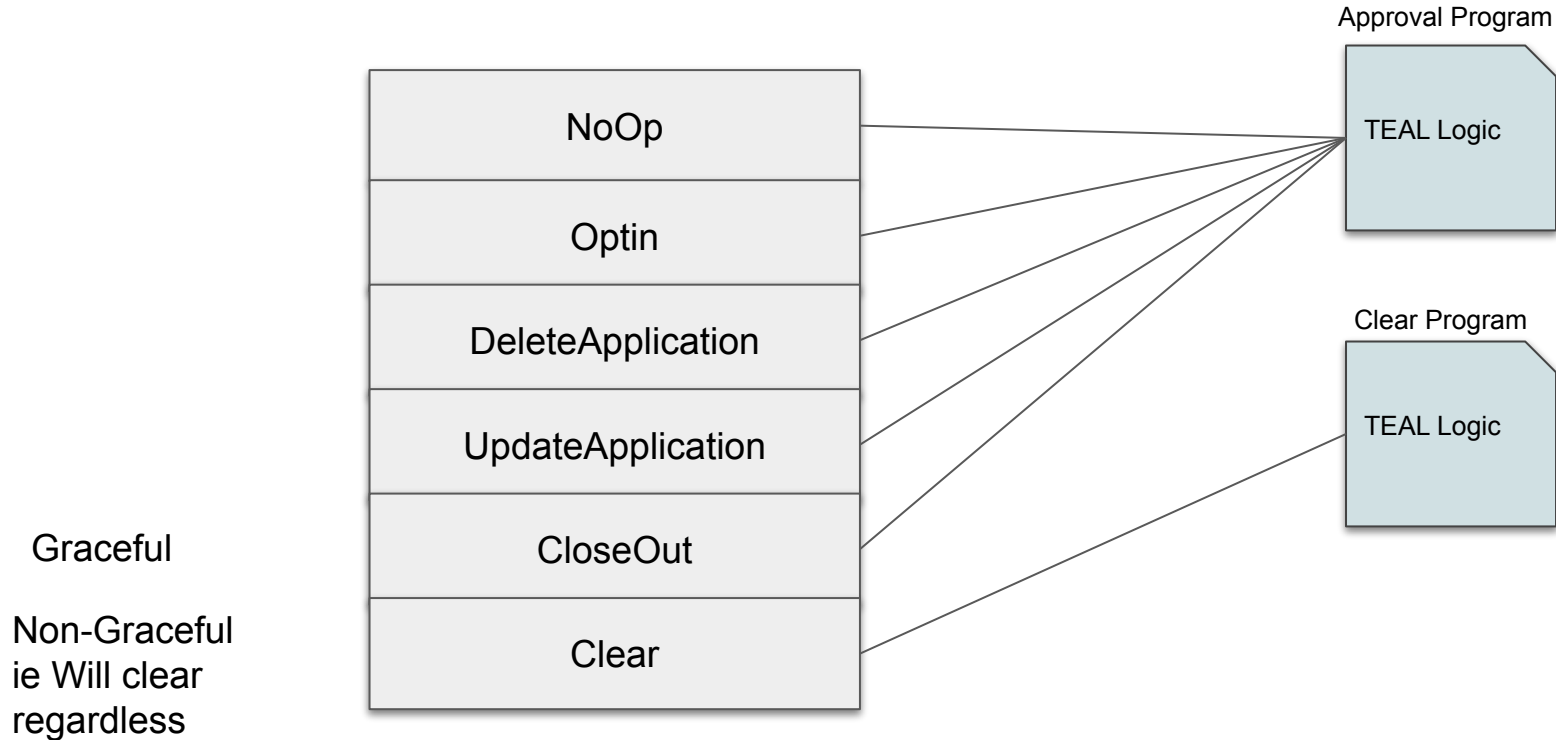


Counter - Step 1

Step 1 - Explanation

Transaction Sub-Types Step 2

Used to Communicate With Smart Contract





Transaction Properties -Types Step 2

Operator	Type	Min TEAL Version	Notes
<code>Txn.sender()</code>	<code>TealType.bytes</code>	2	32 byte address
<code>Txn.fee()</code>	<code>TealType.uint64</code>	2	in microAlgos
<code>Txn.first_valid()</code>	<code>TealType.uint64</code>	2	round number
<code>Txn.last_valid()</code>	<code>TealType.uint64</code>	2	round number
<code>Txn.note()</code>	<code>TealType.bytes</code>	2	transaction note in
<code>Txn.lease()</code>	<code>TealType.bytes</code>	2	transaction lease in
<code>Txn.receiver()</code>	<code>TealType.bytes</code>	2	32 byte address
<code>Txn.amount()</code>	<code>TealType.uint64</code>	2	in microAlgos
<code>Txn.close_remainder_to()</code>	<code>TealType.bytes</code>	2	32 byte address

https://pyteal.readthedocs.io/en/stable/accessing_transaction_field.html#id1



Counter - Step 2 - Transaction Routing

```
Cond([test-expr-1, body-1],  
     [test-expr-2, body-2],  
     . . . )
```

First Success Breaks Out Of Cond()

```
handle_creation = Approve()
```

```
handle_optin = Reject()
```

```
program = Cond(  
    [Txn.application_id() == Int(0), handle_creation],  
    [Txn.on_completion() == OnComplete.OptIn, handle_optin],  
)
```



- OnComplete.OptIn
- OnComplete.CloseOut
- OnComplete.DeleteApplication
- OnComplete.UpdateApplication
- OnComplete.NoOp



Counter - Step 2

5 Min Work Time



Counter - Step 3 - Create and Initialize the Global counter to 0

```
Seq([
    App.globalPut(Bytes("creator"), Txn.sender()),
    Return(Int(1))
])
```

All Expressions But The Last Must
Resolve to Teal Type of None

Not Allowed
Seq(Int(1), Int(1))

```
handle_creation = Seq(
    App.globalPut(Bytes("creator"), Txn.sender()),
    Approve()
)
```

```
App.globalPut(Bytes("status"), Bytes("active")) # write a byte
App.globalPut(Bytes("total supply"), Int(100)) # write a uint64
```

Only Byte[] or UINT64 Allowed

https://pyteal.readthedocs.io/en/stable/control_structures.html#chaining-expressions-seq

<https://pyteal.readthedocs.io/en/stable/state.html#writing-global-state>

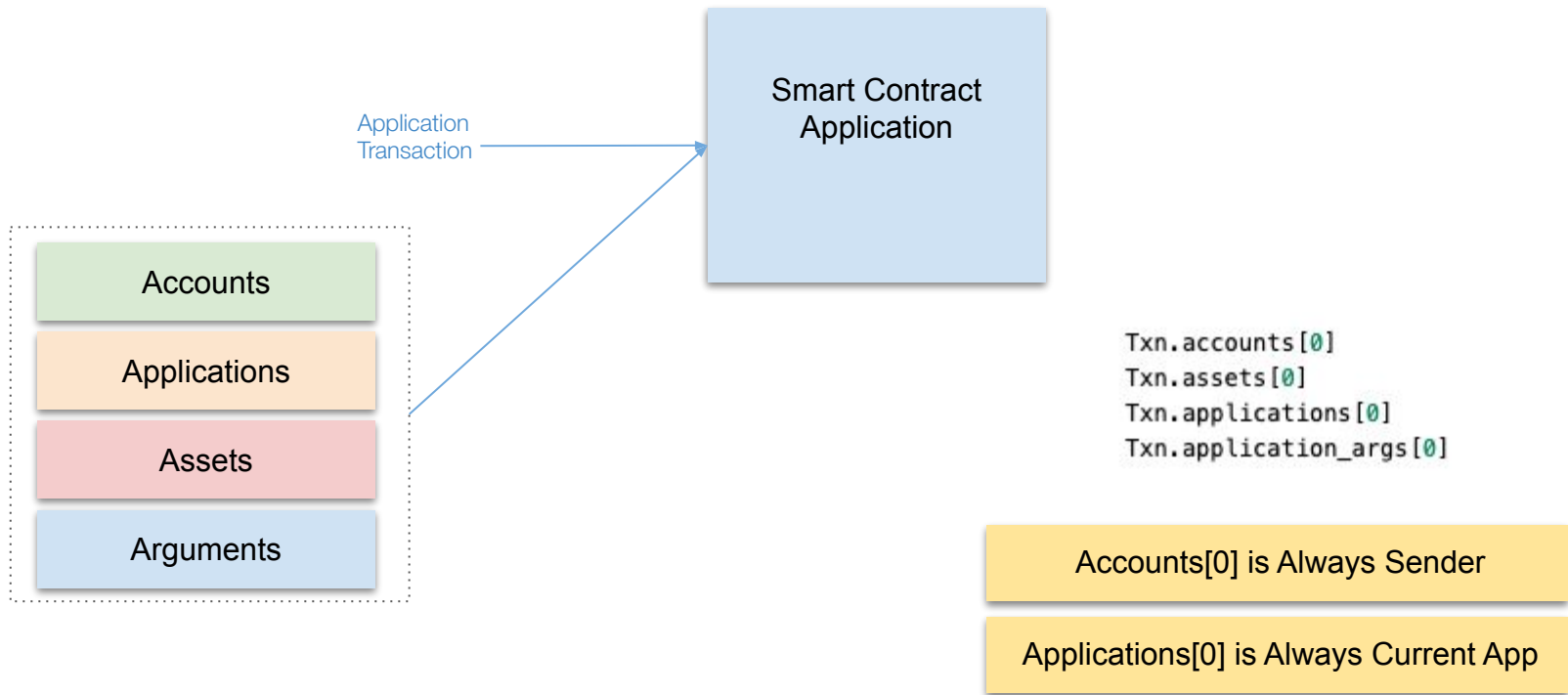


Counter - Step 3

5 Min Work Time



Counter - Step 4 - Arrays





Counter - Step 4 - Get Transaction Parameters, look for Add and Deduct

```
Txn.application_args.length() # get the number of application arguments in the transaction
Txn.application_args[0] # get the first application argument
Txn.application_args[1] # get the second application argument
```

Mod(a, b)	<code>a % b</code>	<code>a % b</code> , modulo operation	<code>Int(7) % Int(3)</code>
Eq(a, b)	<code>a == b</code>	1 if <code>a</code> equals <code>b</code> , 0 otherwise	<code>Int(7) == Int(7)</code>
Neq(a, b)	<code>a != b</code>	0 if <code>a</code> equals <code>b</code> , 1 otherwise	<code>Int(7) != Int(7)</code>

```
add = Approve()
```

```
deduct = Approve()
```

```
handle_noop = Cond(
    [Txn.application_args[0] == ...],
```

Pyteal Overloads Python Operators

https://pyteal.readthedocs.io/en/stable/accessing_transaction_field.html?#transaction-array-fields

https://pyteal.readthedocs.io/en/stable/arithmetic_expression.html



Counter - Step 4

5 Min Work Time



Counter - Step 5 - Implement Add and Deduct

```
App.globalGet(Bytes("status"))  
App.globalGet(Bytes("total supply"))
```

Do not let the Counter go below 0

```
If(test-expr, then-expr)
```

```
If( test-expr, then-expr, else-expr )
```

```
add = Seq(  
  App.globalPut(Bytes("Count"), ....),  
  Approve()  
)  
  
deduct = Seq(  
  If(....,  
    App.globalPut(Bytes("Count"), ...),  
  ),  
  Approve()  
)
```

<https://pyteal.readthedocs.io/en/stable/state.html#reading-global-state>

https://pyteal.readthedocs.io/en/stable/control_structures.html#simple-branching-if



Counter - Step 5

5 Min Work Time



Learn More

Algorand

Pyteal Auction Example

<https://developer.algorand.org/docs/get-started/dapps/pyteal/>

<https://developer.algorand.org/docs/get-details/dapps/pyteal/>

<https://github.com/algorand-devrel/>

- **Discord:** <https://discord.com/invite/84AActu3at>
- Developer Portal (Documentation and Tutorials):
<https://developer.algorand.org/>
- Forum: <https://forum.algorand.org/>
- GitHub: <https://github.com/algorand>
- OFFICE HOURS sign up:
<https://www.algorand.com/developers>