Understanding Algorand's Smart Contract Language

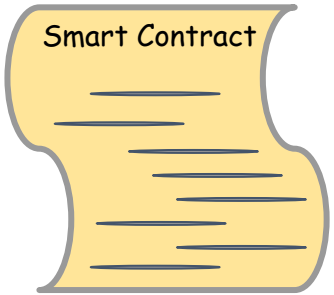
Algorand

Algorand Smart Contracts

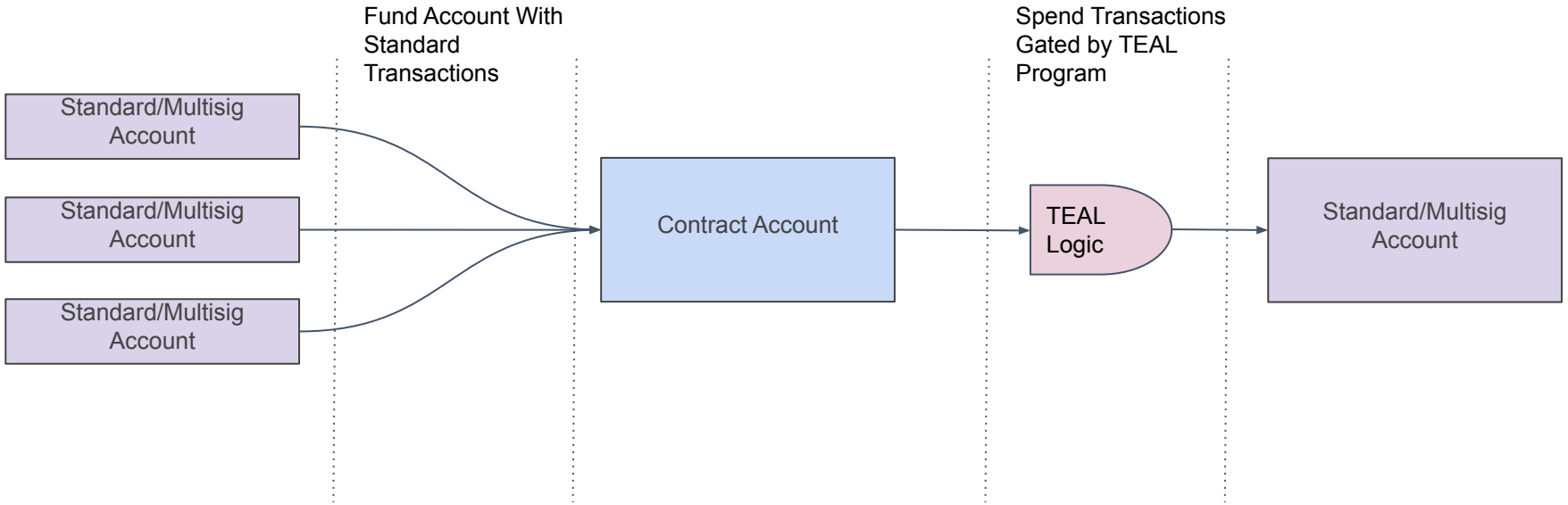
- **Transaction Execution Approval Language**
  - The contract logic on Algorand are described with TEAL
  - Python Enabled Compiler (PyTEAL)

- **Two Types Of Smart Contracts**
  - Stateless - Used to Approve Spending Transactions
  - Stateful - Onchain Global and Local Storage

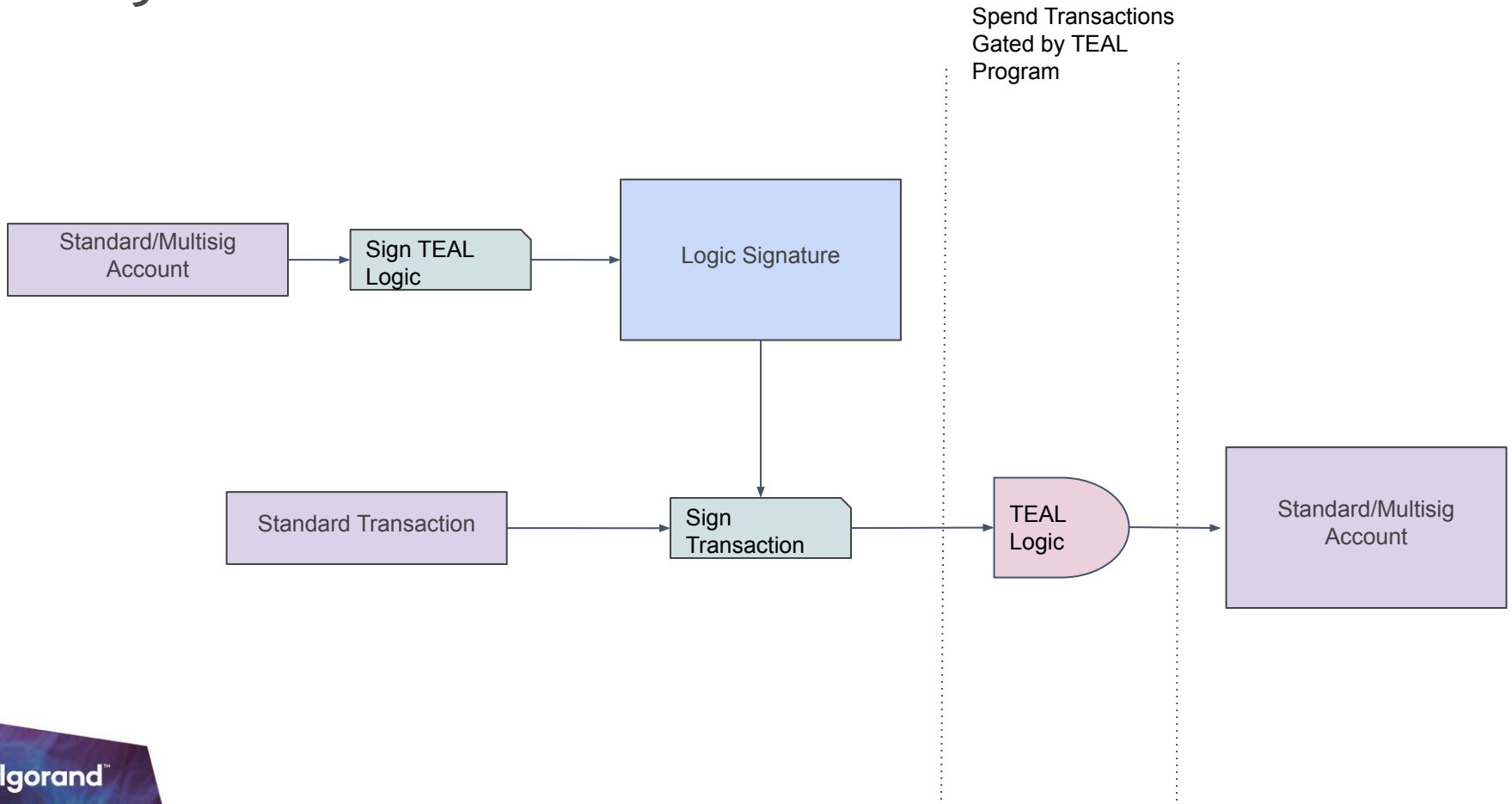
- Combinable with Other Algorand Technology
  - Atomic Transfers
  - Algorand Assets
  - Combine Stateless and Stateful Contracts



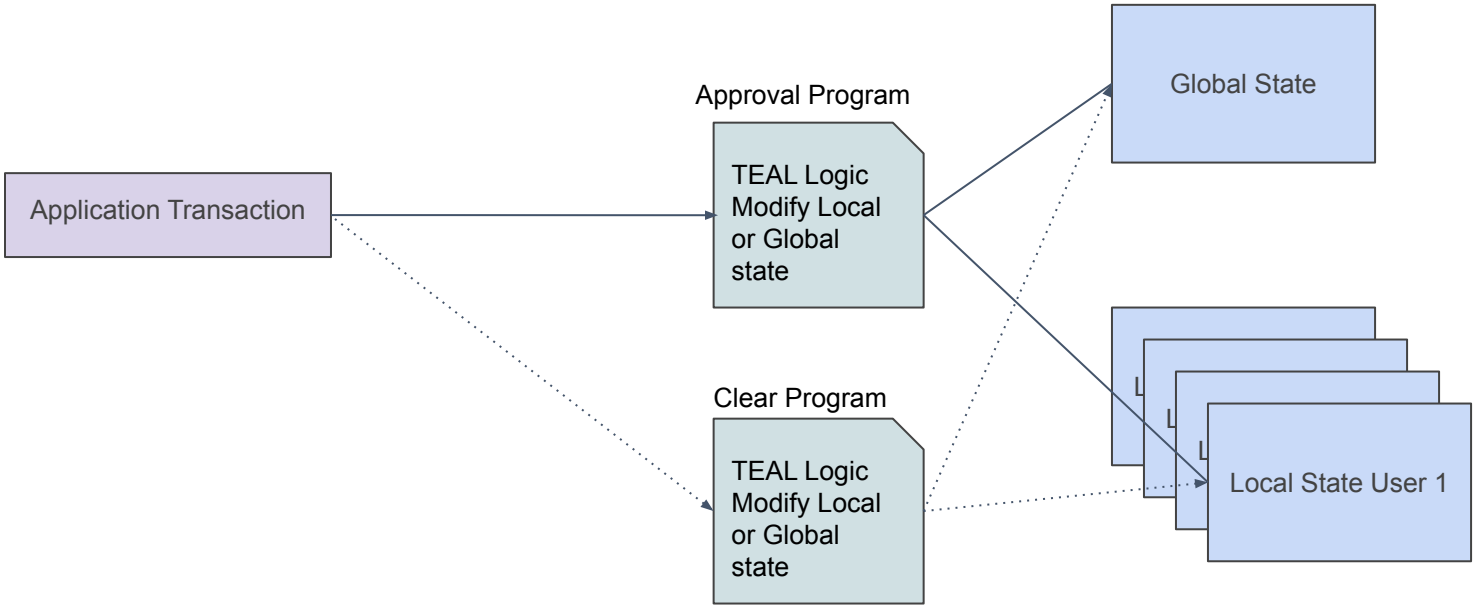
Escrow Stateless



Delegate Stateless

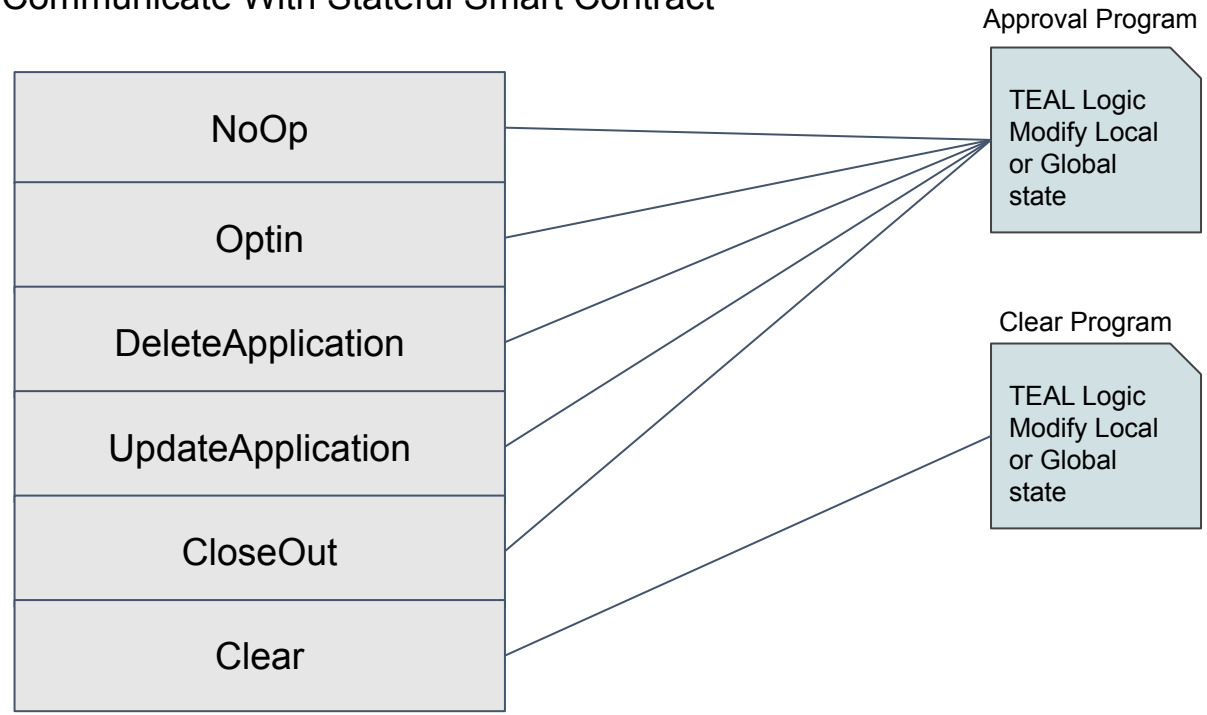


Stateful


Application Transaction
Approval Program
TEAL Logic
Modify Local
or Global
state
Clear Program
TEAL Logic
Modify Local
or Global
state
Global State
Local State User 1

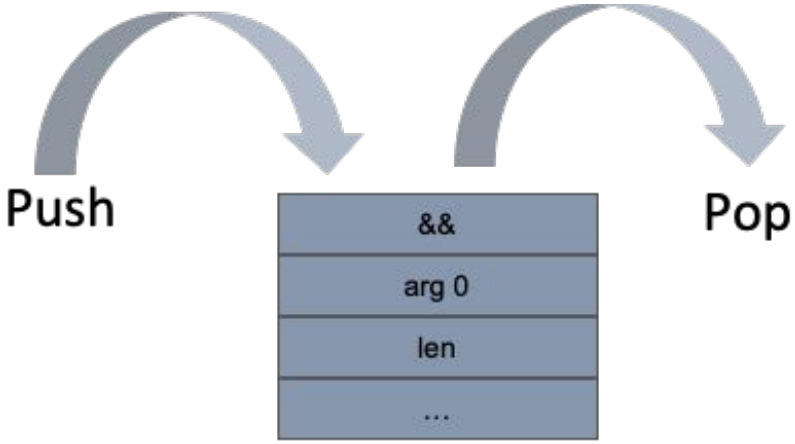
New Transaction Sub-Types for Application

Used to Communicate With Stateful Smart Contract

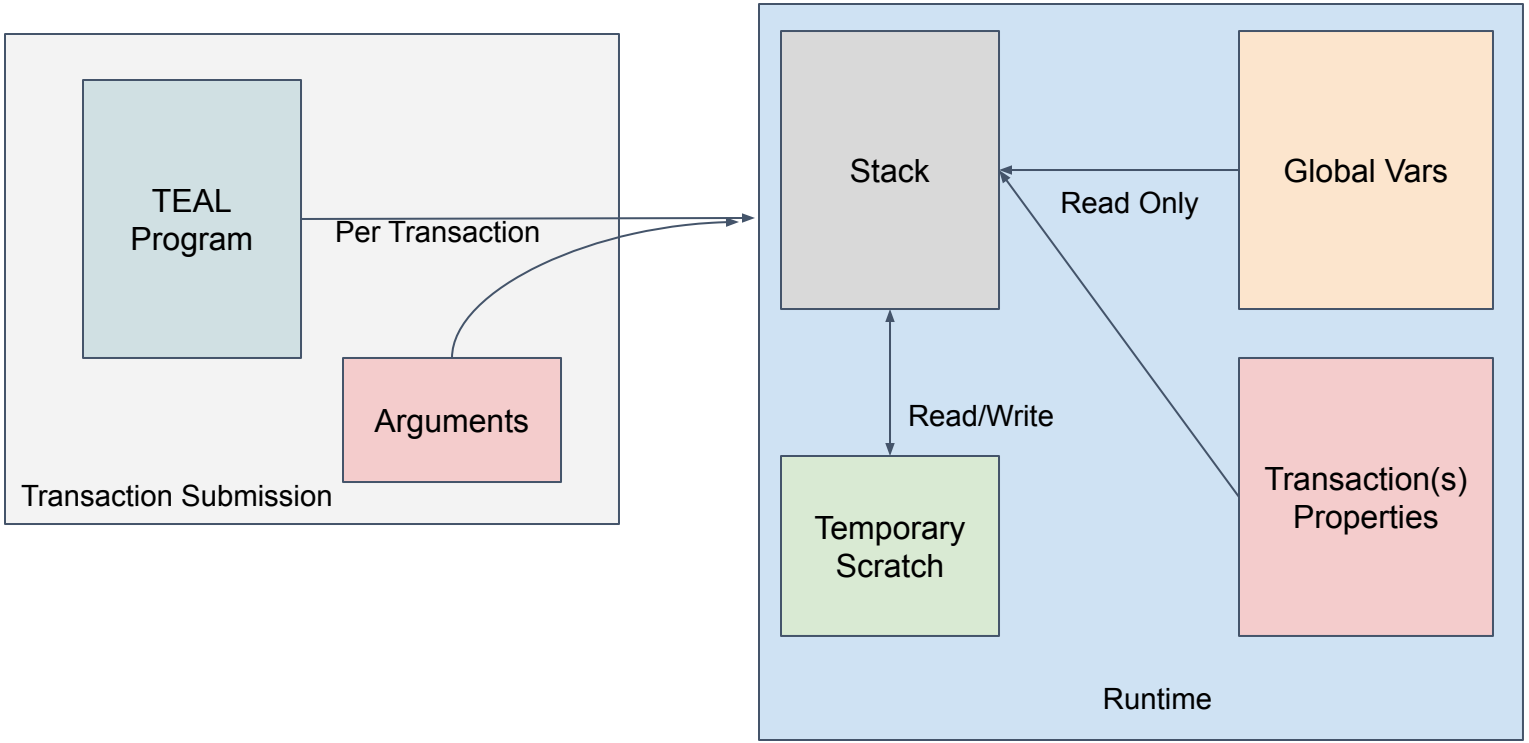


TEAL - Transaction Execution Approval Language

- Bytecode based stack language
- Returns True or False (One Positive Value Left on Stack)
- SDK Support
- > 70 Opcodes
- Access to ASA/Algo Balances
- Read all Transactions in a Group
- Stateful - Global/Local Storage Calls
- Stateless - Signature Verification
- PyTEAL library to write in python



Stateless Runtime Architecture



Teal Stack Architecture (Stateless)

Program

```
txn CloseRemainderTo
addr SOEI...
==
txn Receiver
addr SOEI...
==
&&
arg 0
len
int 32
==
&&
arg 0
sha256
byte base64 VeU...
==
&&
txn CloseRemainderTo
addr RFGE...
==
...
```

Stack

uint64/[]byte
uint64/[]byte
uint64/[]byte

…(up to 1000)

Scratch Space

0: uint64/[]byte
1: uint64/[]byte
2: uint64/[]byte
...
255: uint64/[]byte

Args

(This txn only)

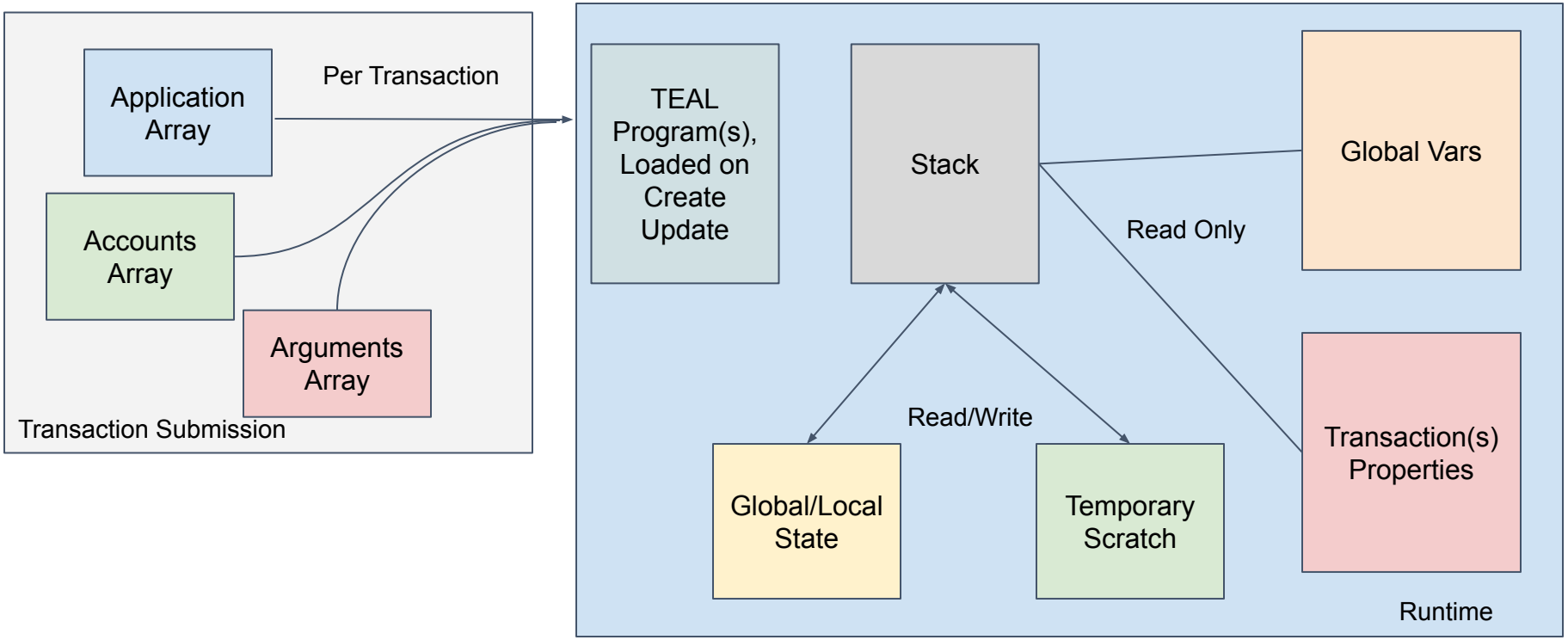
0: []byte
1: []byte
2: []byte

…(up to 255)

Transaction(s)

- Sender
- Fee
- FirstValid
- FirstValidTime
- LastValid
- Note
- Lease
- Receiver
- Amount
- CloseRemainderTo
- VotePK
- SelectionPK
- VoteFirst
- VoteLast
- VoteKeyDilution
- Type
- TypeEnum
- XferAsset
- AssetAmount
- AssetSender
- AssetReceiver
- AssetCloseTo
- GroupIndex
- TxID

Stateful Runtime Architecture



Teal Stack Architecture (Stateful)

Program

```
txn CloseRemainderTo
addr SOEI...
==
txn Receiver
addr SOEI...
==
&&
arg 0
len
int 32
==
&&
arg 0
sha256
byte base64 VeU...
==
&&
txn CloseRemainderTo
addr RFGE...
==
...
```

Stack

uint64/[]byte
uint64/[]byte
uint64/[]byte

…(up to 1000)

Arguments*

[]byte
[]byte
[]byte

…(up to 16 & 2K)

Accounts*

[]byte
[]byte

…(up to 4 accounts)

Scratch Space

0: uint64/[]byte
1: uint64/[]byte
2: uint64/[]byte
...
255: uint64/[]byte

Applications*

[]byte
[]byte

…(up to 2 app ids)

Global State 64 K/V pairs 64 bytes

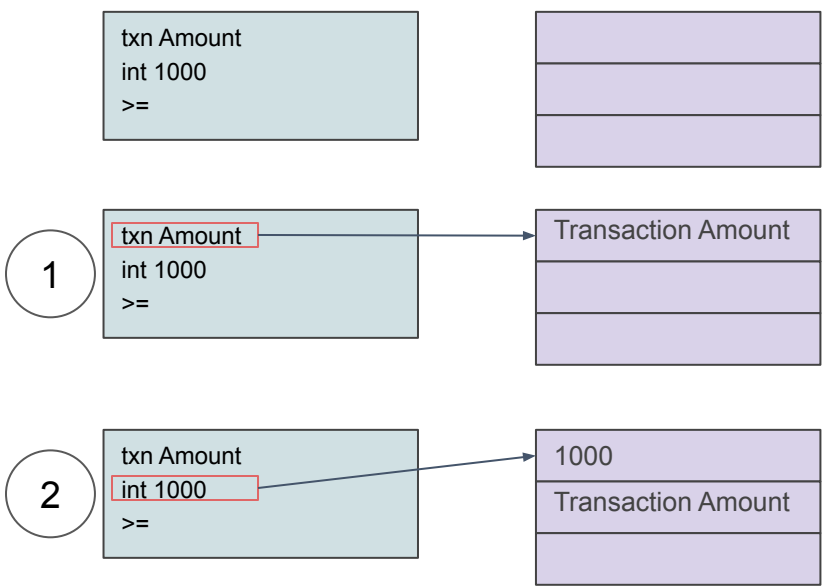
Local State 16 K/V pairs 64 bytes

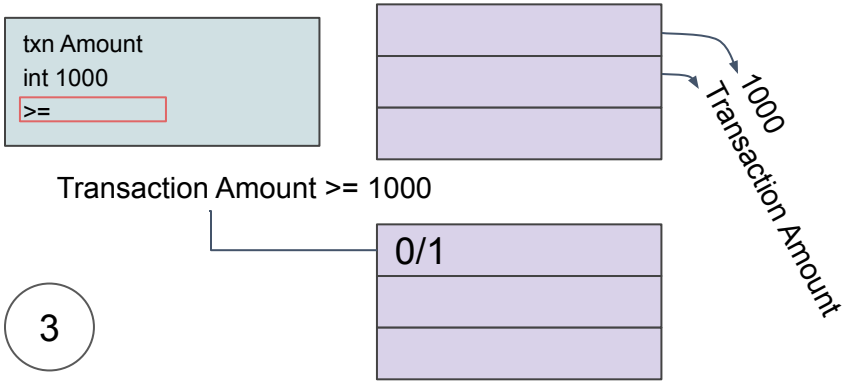
* Can change per transaction

Transaction(s)

- Sender
- Fee
- FirstValid
- FirstValidTime
- LastValid
- Note
- Lease
- Receiver
- Amount
- CloseRemainderTo
- VotePK
- SelectionPK
- VoteFirst
- VoteLast
- VoteKeyDilution
- Type
- TypeEnum
- XferAsset
- AssetAmount
- AssetSender
- AssetReceiver
- AssetCloseTo
- GroupIndex
- TxID

Simple Stack Example





>=

- Opcode: 0x0f
- Pops: ... *stack*, {uint64 A}, {uint64 B}
- Pushes: uint64
- A greater than or equal to B => {0 or 1}

Opcodes

app_local_get

- Opcode: 0x62
- Pops: ... *stack*, {uint64 A}, {[]byte B}
- Pushes: any
- read from account specified by Txn.Accounts[A] from local state of the current application key B => value
- LogicSigVersion >= 2
- Mode: Application

params: account index, state key. Return: value. The value is zero if the key does not exist.

Opcode Reference Document

Pseudo Operators

int //load an int onto stack

byte //Load bytes on stack

addr //Load Algorand address

TEAL Approval Program

```
txn Receiver
addr HJLWACXDBOEH25KJB2WI2X5BHQOJ4LS2MRLNMHJ5ZZOBLJU7KGDJSHEU3I
==
.
byte “mystring”
txn ApplicationArgs 0
==
```

Accessing Transaction Properties

- Sender
- Fee
- FirstValid
- FirstValidTime
- LastValid
- Note
- Lease
- Receiver
- Amount
- CloseRemainderTo
- VotePK
- SelectionPK
- VoteFirst
- VoteLast
- VoteKeyDilution
- Type
- TypeEnum
- XferAsset
- AssetAmount
- AssetSender
- AssetReceiver
- AssetCloseTo
- GroupIndex
- TxID

TEAL Approval Program

```
txn Amount
int 1000
>=
```

State Demo

Branching Demo

Global Variables

Index	Name	Type	Notes
0	MinTxnFee	uint64	micro Algos
1	MinBalance	uint64	micro Algos
2	MaxTxnLife	uint64	rounds
3	ZeroAddress	[]byte	32 byte address of all zero bytes
4	GroupSize	uint64	Number of transactions in this atomic transaction group. At least 1
5	LogicSigVersion	uint64	Maximum supported TEAL version. LogicSigVersion >= 2.
6	Round	uint64	Current round number. LogicSigVersion >= 2.
7	LatestTimestamp	uint64	Last confirmed block UNIX timestamp. Fails if negative. LogicSigVersion >= 2.
8	CurrentApplicationID	uint64	ID of current application executing. Fails if no such application is executing. LogicSigVersion >= 2.

Checking Type of Transaction

Value	Constant name	Description
0	unknown	Unknown type. Invalid transaction
1	pay	Payment
2	keyreg	KeyRegistration
3	acfg	AssetConfig
4	axfer	AssetTransfer
5	afrz	AssetFreeze
6	appl	ApplicationCall

TEAL Approval Program

```
txn TypeEnum
int appl
==
```

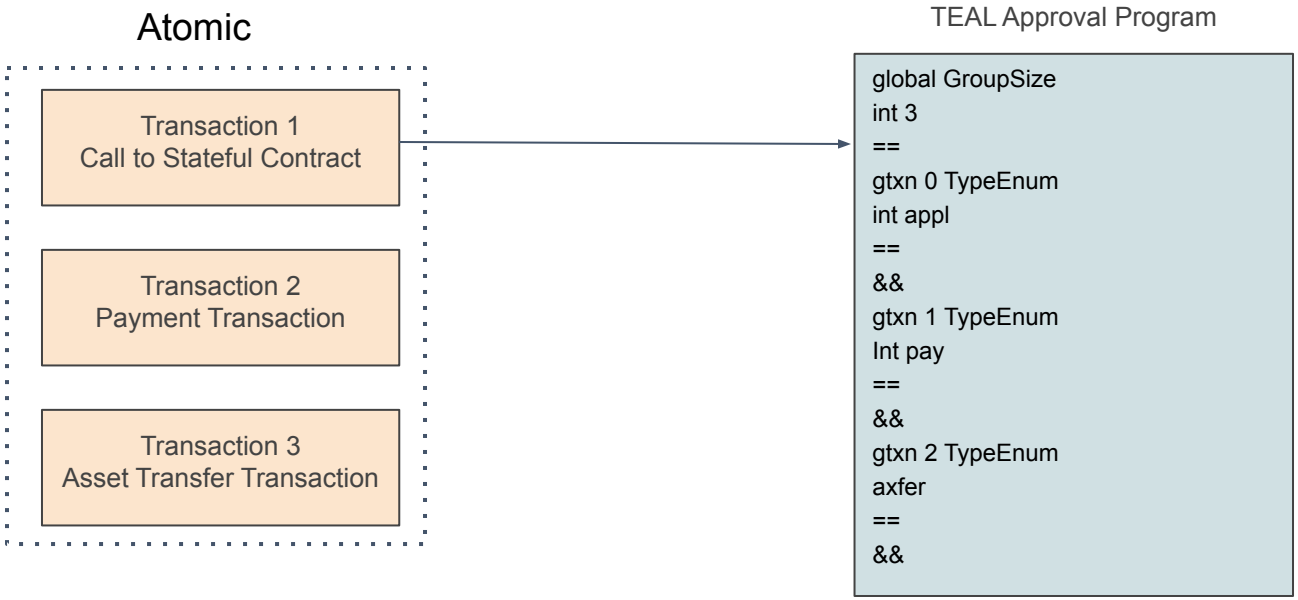
Application Transaction Sub-Types - Stateful

0	NoOp	Only execute the `ApprovalProgram` associated with this application ID, with no additional effects.
1	OptIn	Before executing the `ApprovalProgram`, allocate local state for this application into the sender's account data.
2	CloseOut	After executing the `ApprovalProgram`, clear any local state for this application out of the sender's account data.
3	ClearState	Don't execute the `ApprovalProgram`, and instead execute the `ClearStateProgram` (which may not reject this transaction). Additionally, clear any local state for this application out of the sender's account data as in `CloseOutOC`.
4	UpdateApplication	After executing the `ApprovalProgram`, replace the `ApprovalProgram` and `ClearStateProgram` associated with this application ID with the programs specified in this transaction.
5	DeleteApplication	After executing the `ApprovalProgram`, delete the application parameters from the account data of the application's creator.

TEAL Approval Program

```
txn OnCompletion
int NoOp
==
```

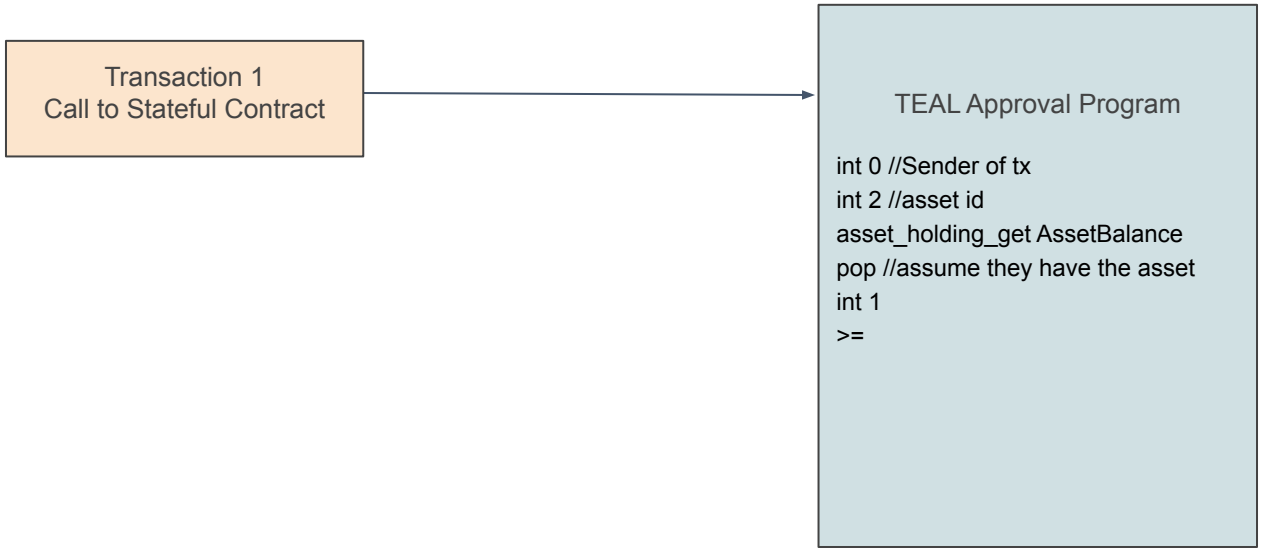
Atomic Transactions - gtxn vs txn



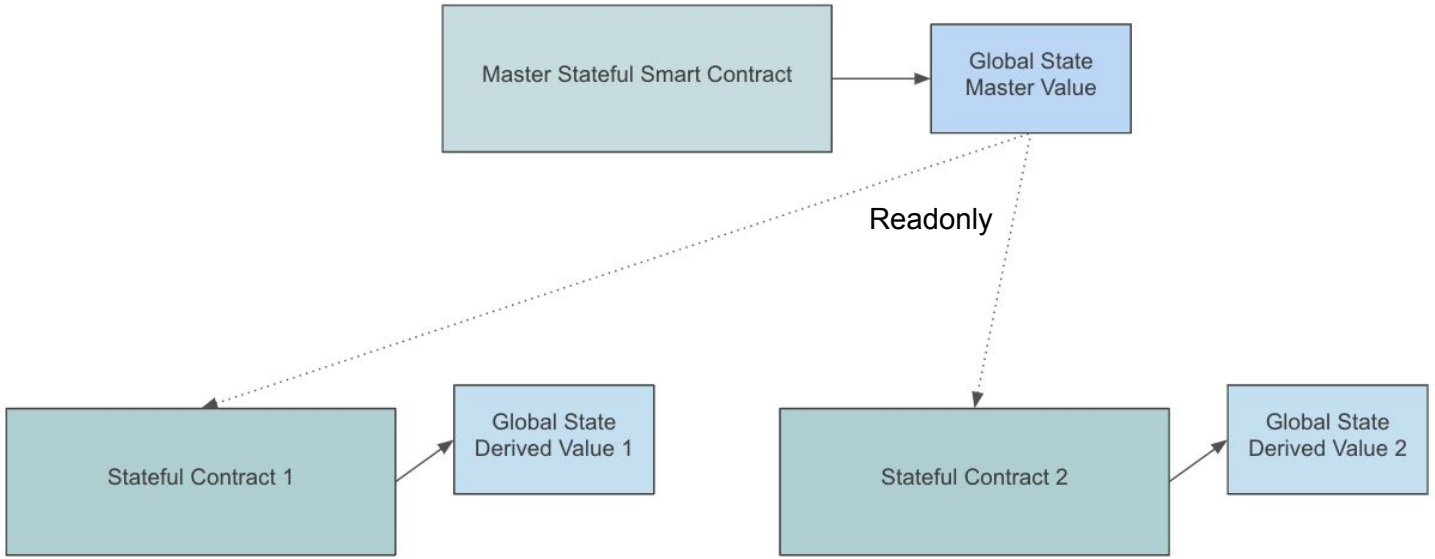
Atomic Demo

Algorand

Asset Check


Transaction 1
Call to Stateful Contract
TEAL Approval Program
int 0 //Sender of tx
int 2 //asset id
asset_holding_get AssetBalance
pop //assume they have the asset
int 1
>=

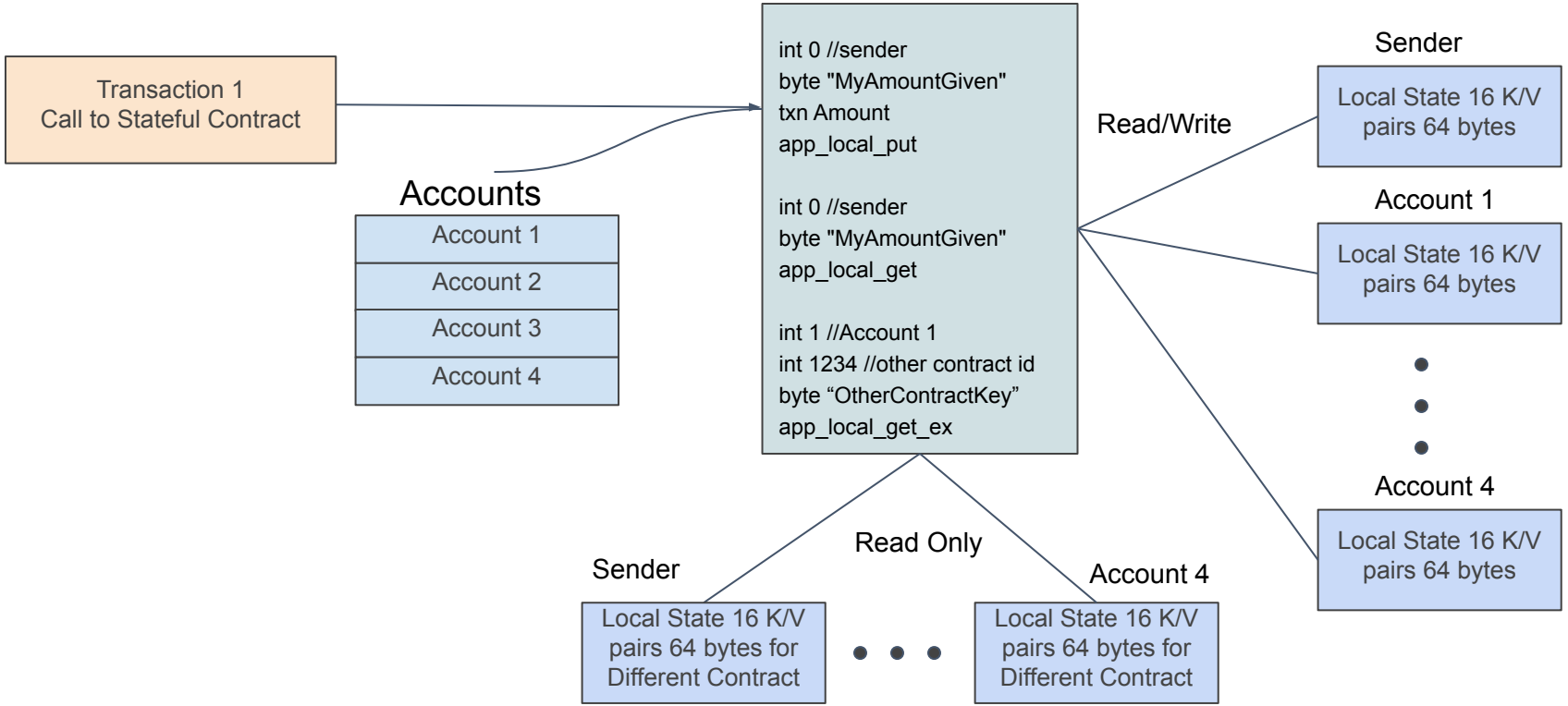
Read Global State From Another Contract



Read Other Contracts State Demo

Algorand

Reading and Writing Local State



Escrow Demo

Algorand

PyTEAL -Python Library

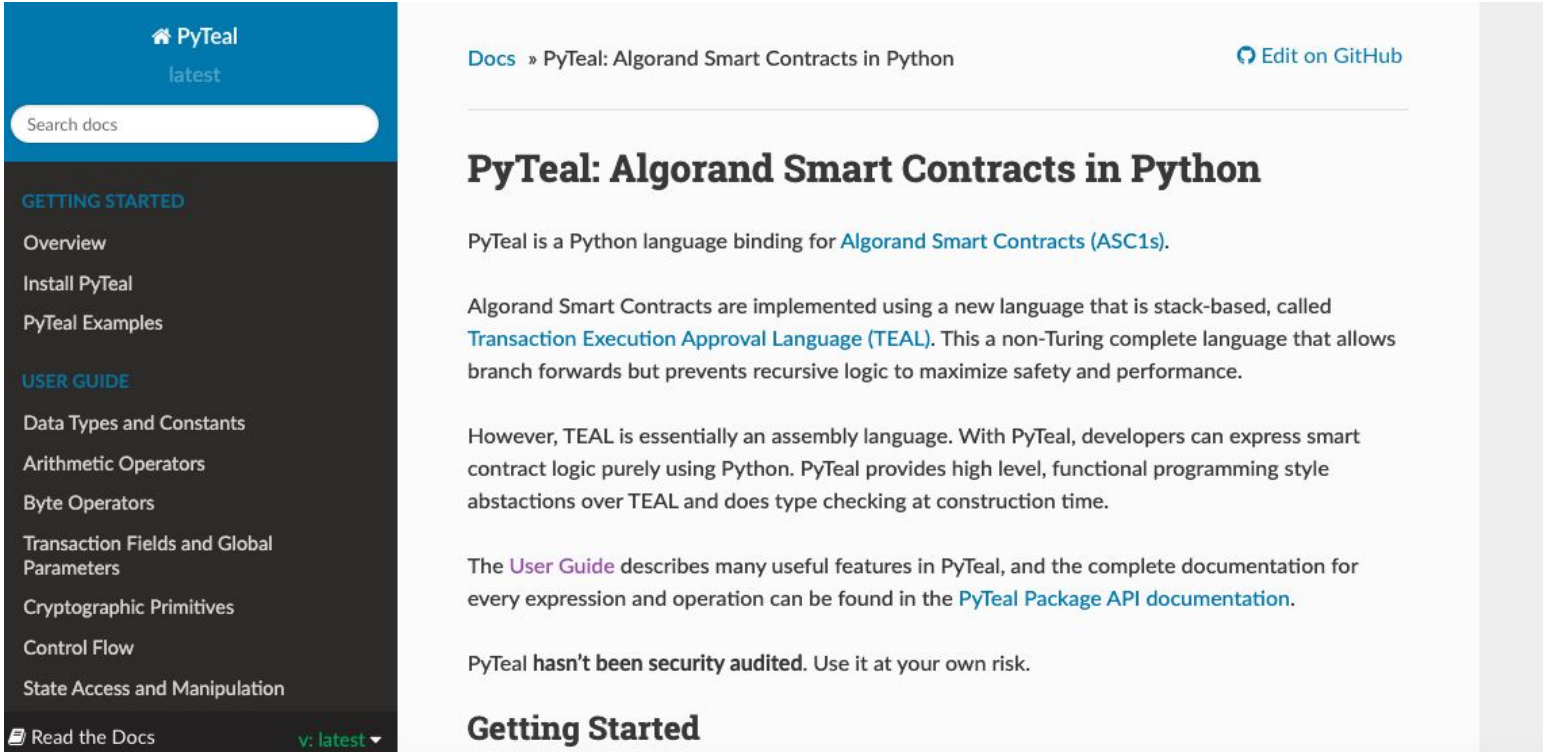
```
def bank_for_account(receiver):
    is_payment = Txn.type_enum() == TxnType.Payment
    is_single_tx = Global.group_size() == Int(1)
    is_correct_receiver = Txn.receiver() == Addr(receiver)

    return And(
        is_payment,
        is_single_tx,
        is_correct_receiver
    )
if __name__ == "__main__":
    program =
bank_for_account("ZZAF5ARA4MEC5PVDOP64JM5O5MQST63Q2KOY2FLYFL
XXD3PFSNJJBYAFZM")
    print(compileTeal(program, Mode.Signature))
```

Returns

```
#pragma version 2
txn TypeEnum
int pay
==
global GroupSize
int 1
==
&&
txn Receiver
addr
ZZAF5ARA4MEC5PVDOP64JM5O5MQST63Q2KOY2FLYFLXXD3P
FSNJJBYAFZM
==
&&
```

PyTeal Documentation



Documentation

PyTeal Demo

Follow Guidelines for Safety

Boilerplate Approval Program

Example Stateless Contract

Adhere to These Guidelines to Protect Accounts

Size Limitations and Opcode Cost Limits

Resources

- **Discord: https://discord.gg/YgPTCVk**
- 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