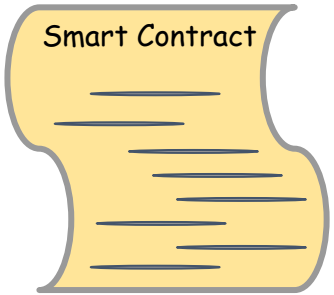
Algorand Smart Contracts
TEAL Version 3 Update

Algorand

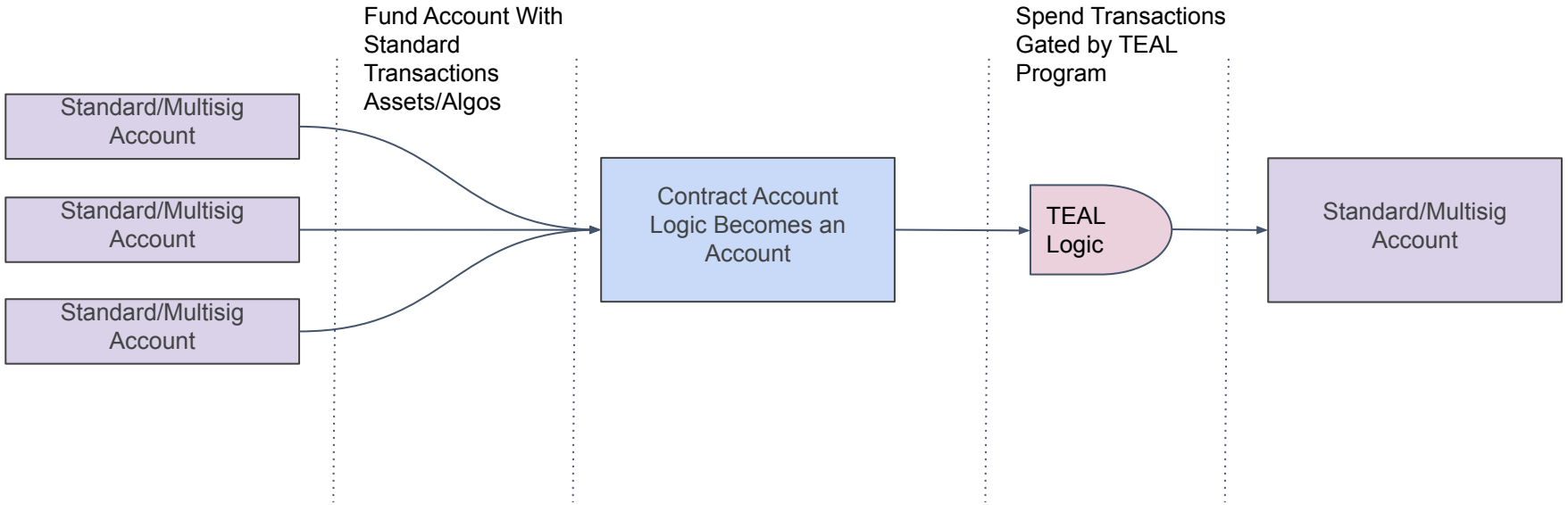
Quick Review

Algorand Smart Contracts

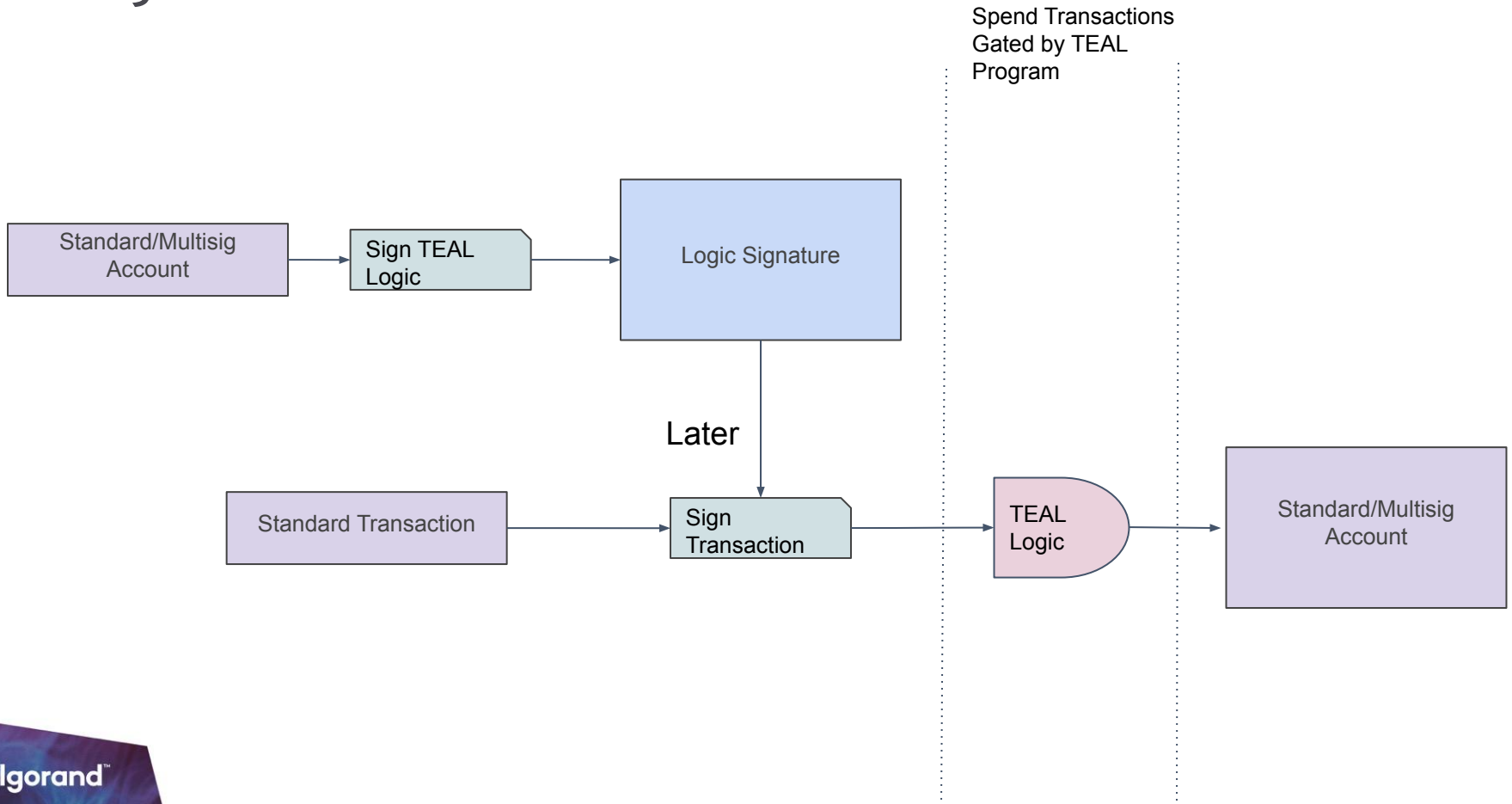
- **Two Types Of Smart Contracts**
  - Stateless - Used to Approve Spending Transactions
    - Contract/Escrow Account
    - Delegate
  - Stateful - Onchain Global and Local Storage
- Combinable with Other Algorand Technology
  - Atomic Transfers
  - Algorand Assets
  - Combine Stateless and Stateful Contracts
- **Transaction Execution Approval Language**
  - The contract logic is written in TEAL
  - Python Enabled Compiler (PyTEAL)
  - Reach Framework for Generating Formally Verified Contracts



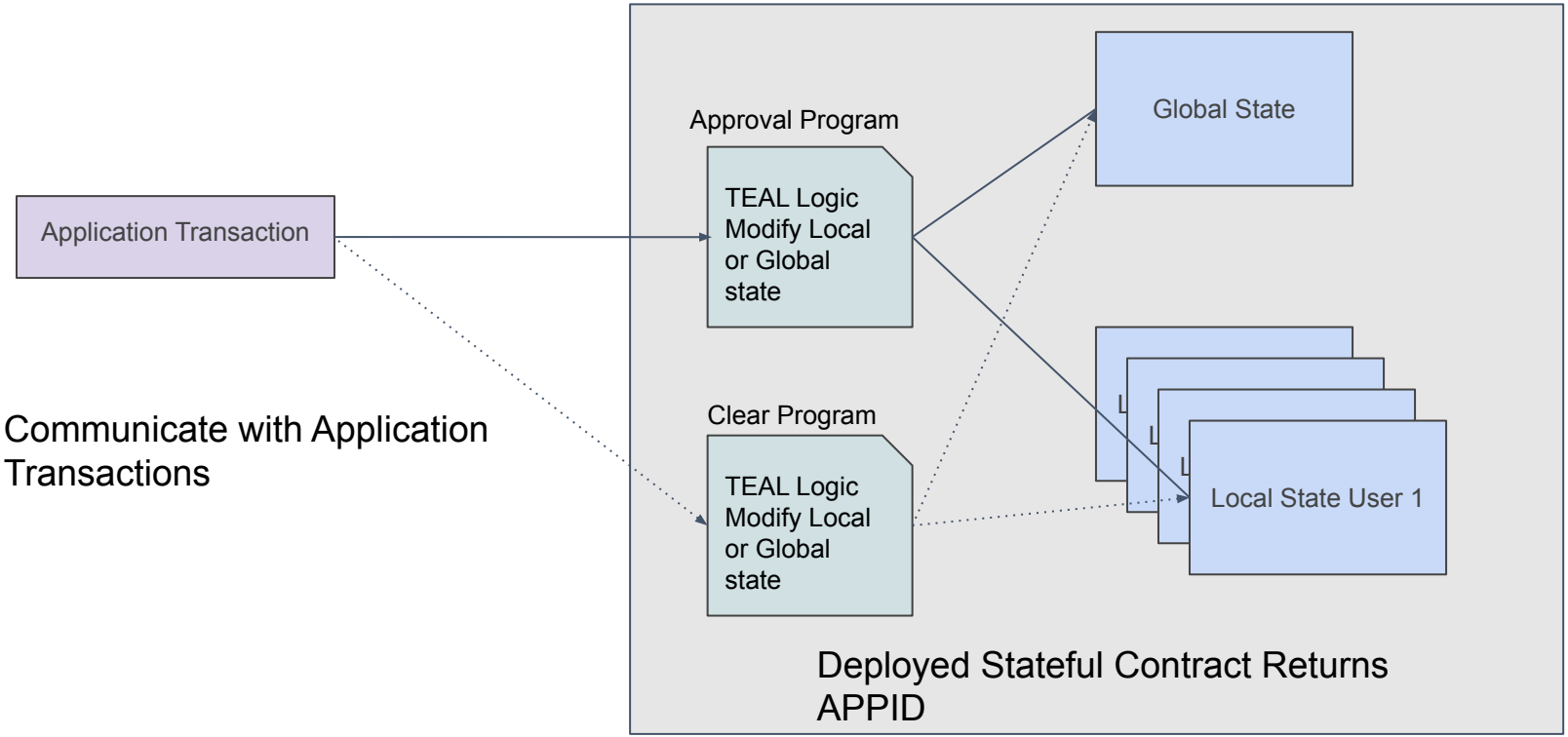
Escrow/Contract Account Stateless



Delegate Stateless



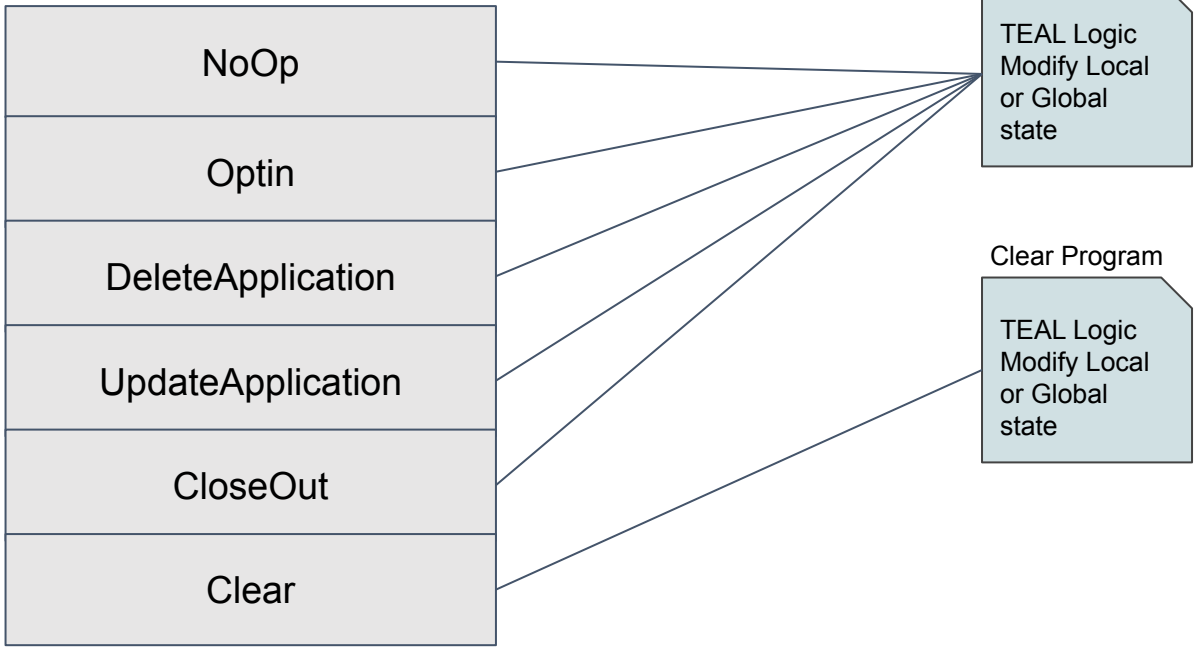
Stateful aka Apps



Communicate with Application Transactions

New Transaction Sub-Types for Application

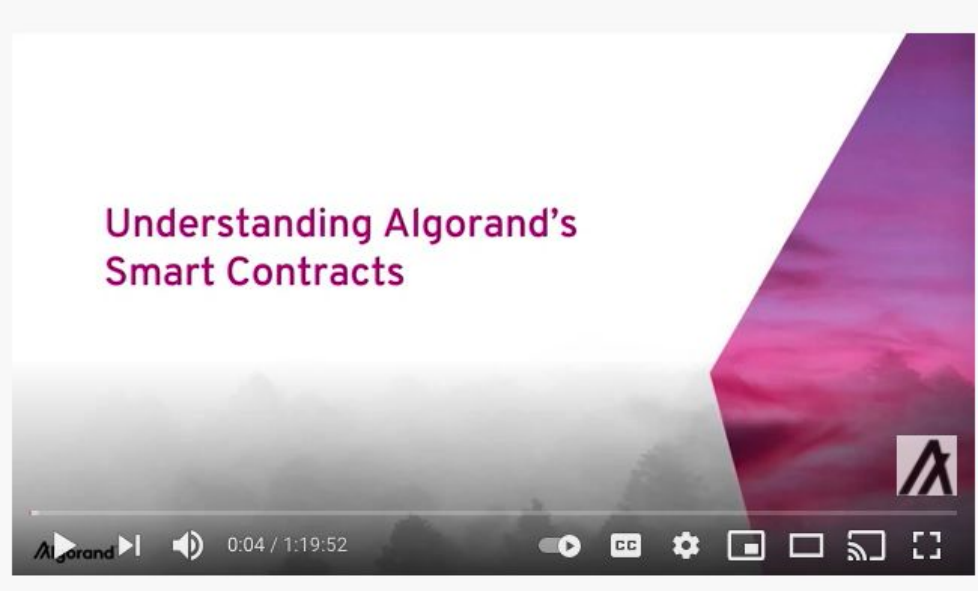
Used to Communicate With Stateful Smart Contract



Graceful

Non-Graceful
ie Will clear
regardless

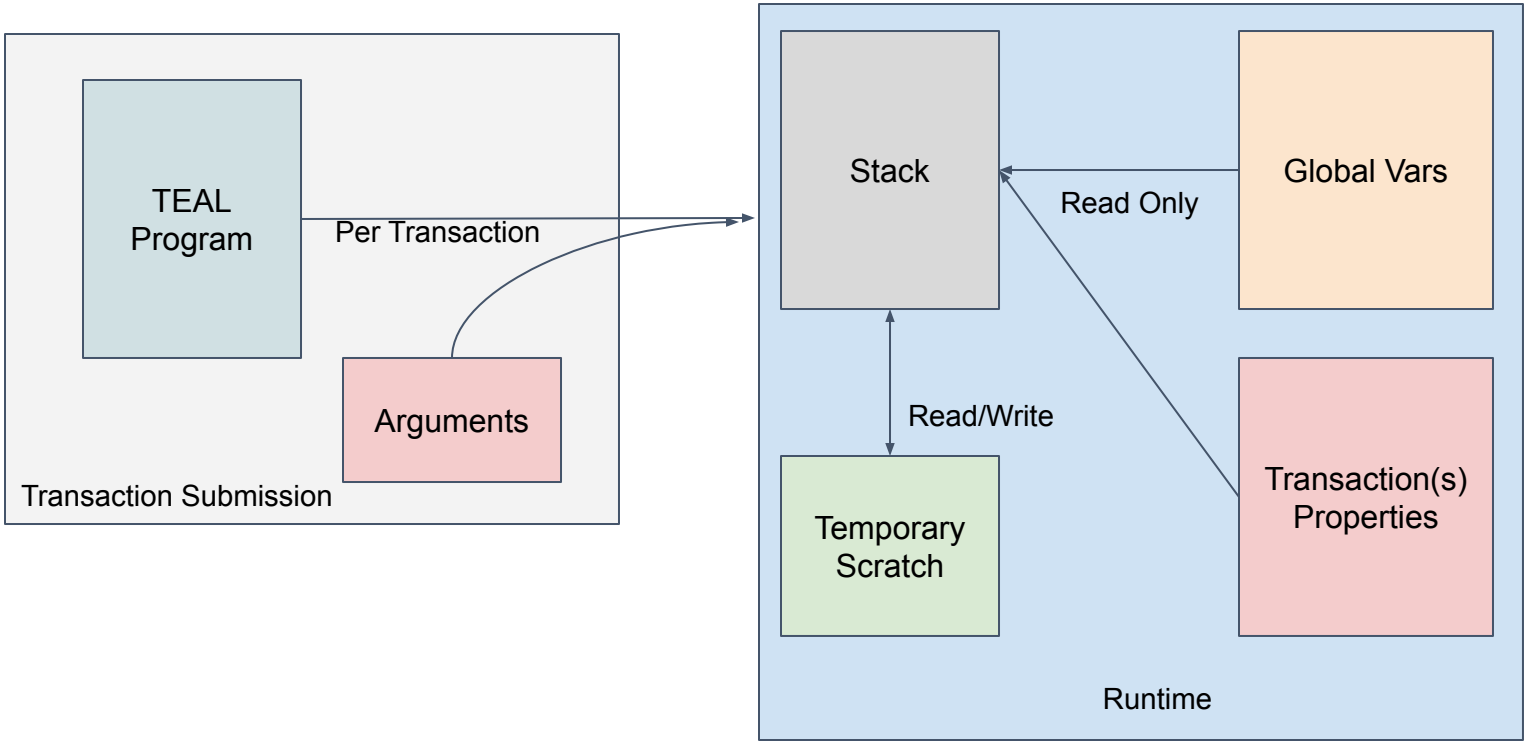
Full Smart Contract Talk



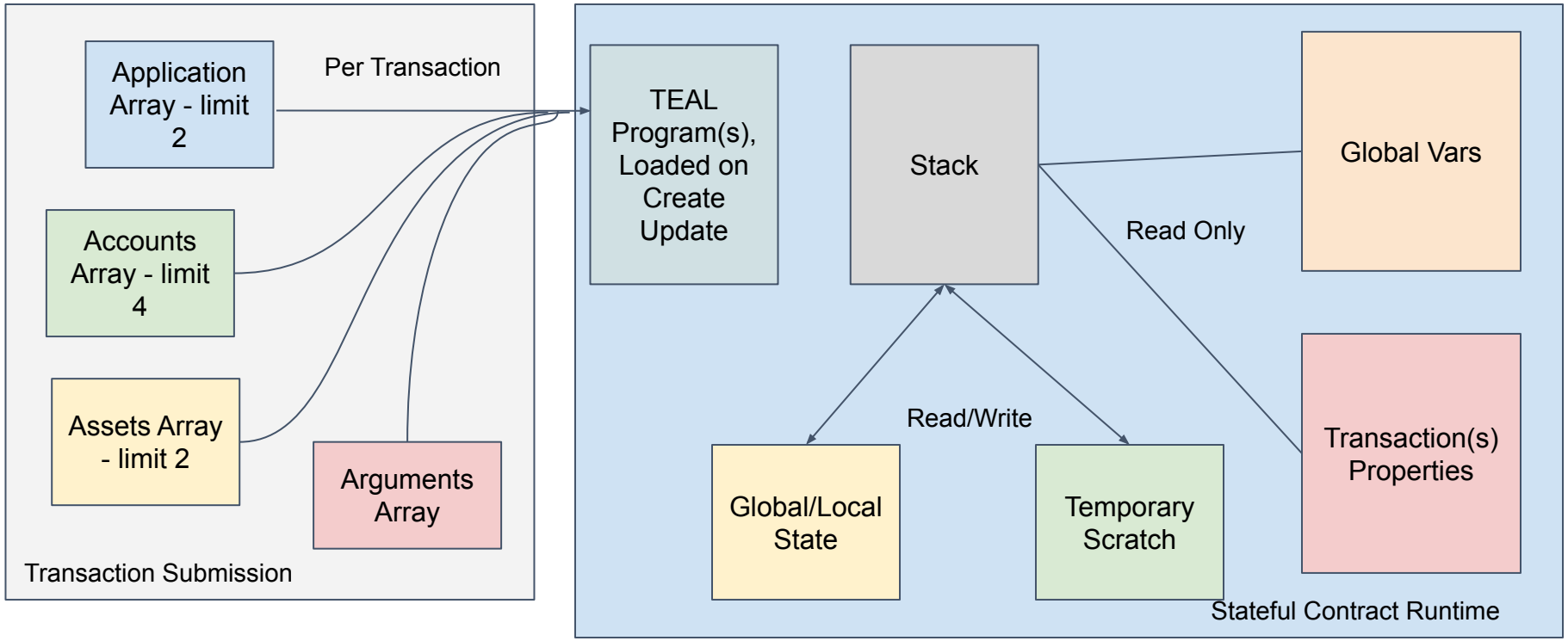
https://www.youtube.com/watch?v=9EpGKexKeMk

Runtime Architecture

Stateless Runtime Architecture



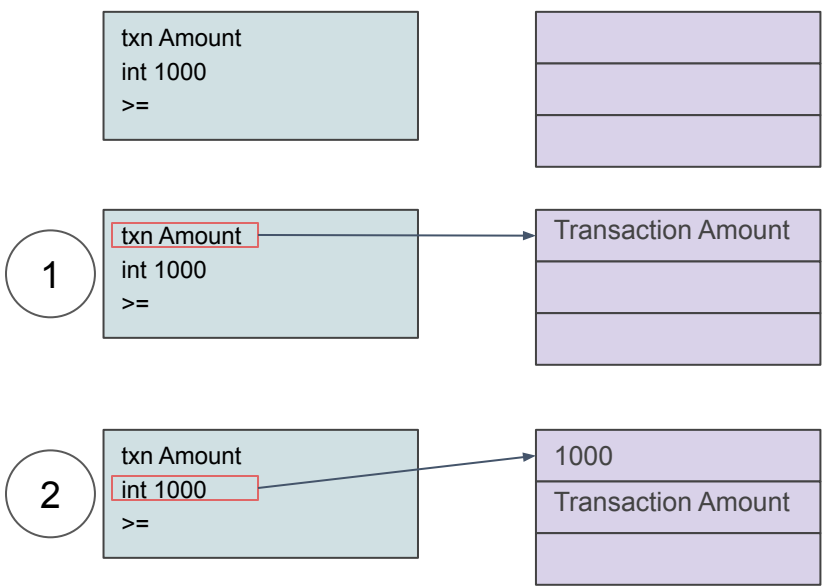
Stateful Runtime Architecture

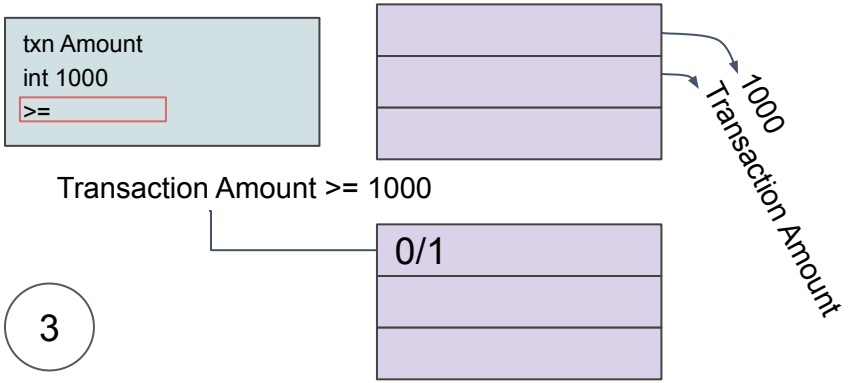


TEAL Version 3 Updates

- Additional Transaction Properties
- New Global for Creator
- New Opcodes
- Group transaction improvements

Simple Stack Example





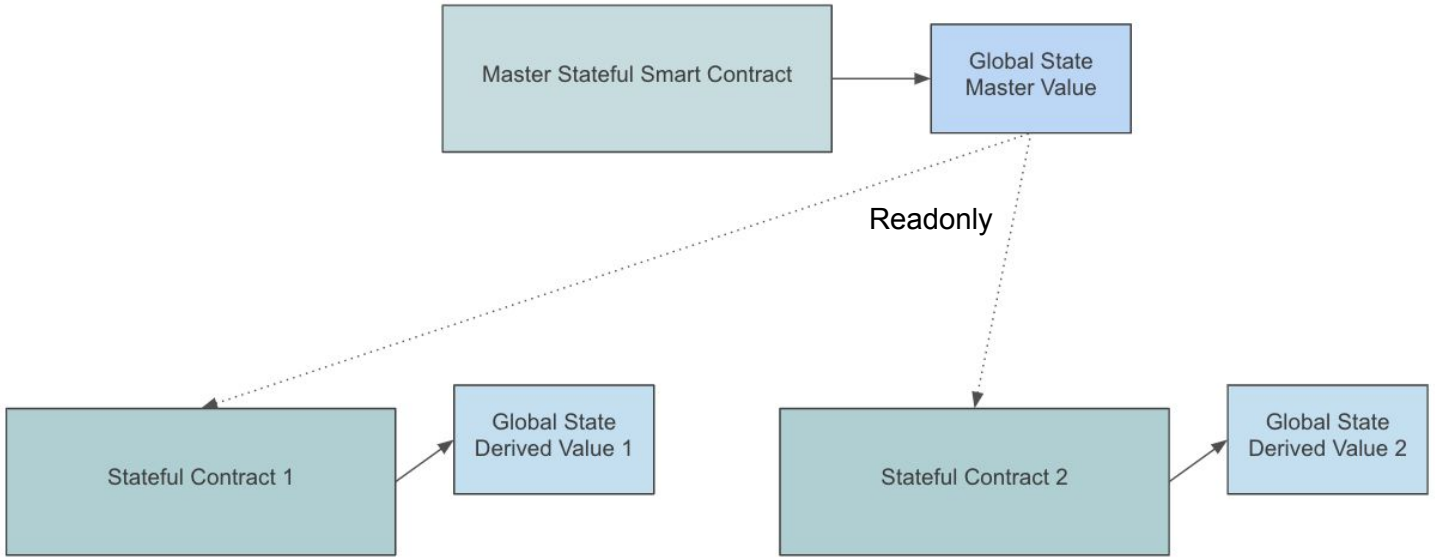
>=

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

TEAL Versioning

- TEAL Version 1
  - `global GroupSize`
  - `int 2`
  - `==`
  - `….`
- TEAL Version 2
  - `#pragma version 2`
  - `global GroupSize`
  - `int 2`
  - `==`
  - `...`
- TEAL Version 3
  - `#pragma version 3`
  - `global GroupSize`
  - `int 2`
  - `==`
  - `...`

Read Global State From Another Contract



Additional Txn Props for Apps

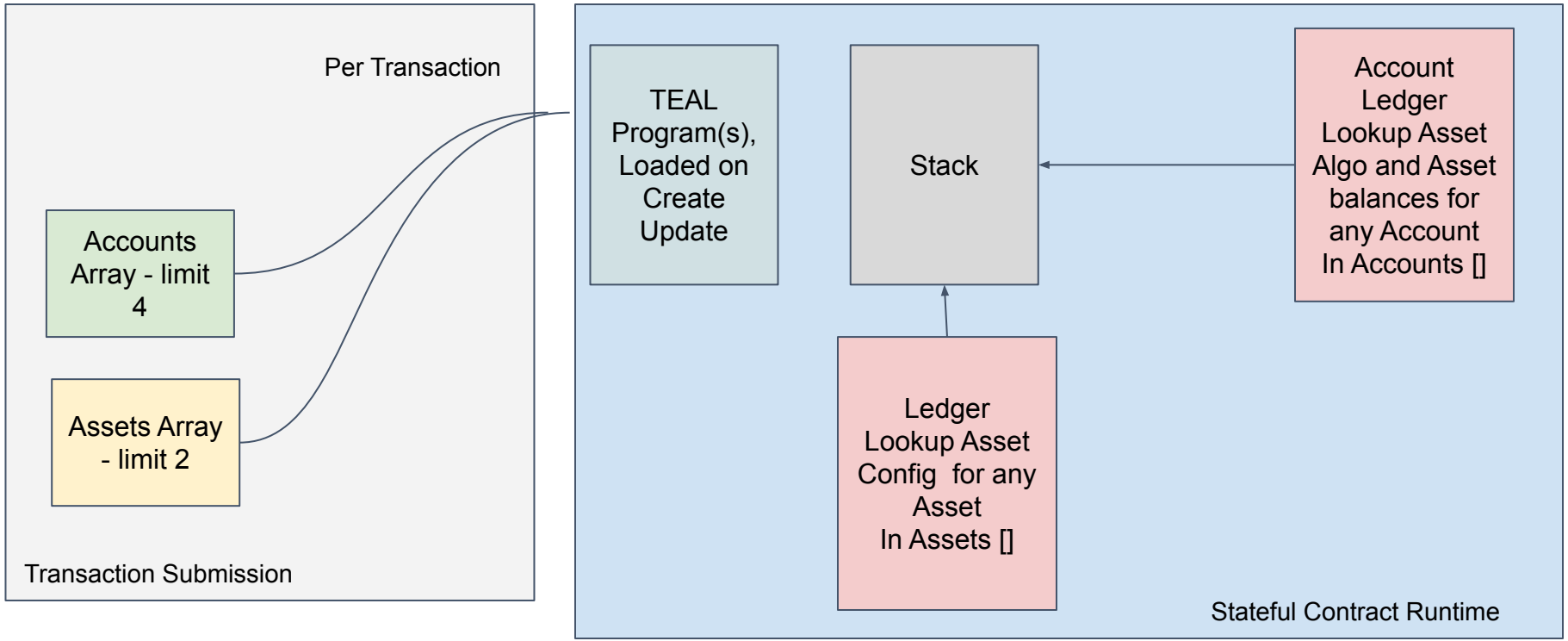
- NumApplications
- Applications

```
// Verify current transaction has two Applications
// passed with the transaction
txn NumApplications
int 2
==

```
// Check that the first foreign Application
// passed in Application ID is = 123456
txn Applications 1
int 123456
```
```

Demo

Stateful Runtime Architecture



Additional Txn Props for Assets

- NumAssets
- Assets

```
// Verify current transaction has two Assets
// passed with the transaction
txn NumAssets
int 2
==
```

```
// Check the second transaction in a group to verify
// it is passed the Asset ID 123456 as the first asset
// in the assets array
gtxn 1 Assets 0
int 123456
==
```

Demo

New Global for Creator Stateful Contracts

- Creator
  - Save Global Space

```
txn Sender
global CreatorAddress
==
```

New Txn Props for Checking Storage Used

- GlobalNumUint
- LocalNumUint
- GlobalNumByteSlice
- LocalNumByteSlice

```
txn GlobalNumUint
int 2
==
txn GlobalNumByteSlice
int 1
==
&&
txn LocalNumUint
int 4
==
&&
txn LocalNumByteSlice
int 3
==
&&
```

Demo

New Opcodes

New Assert Opcode

- assert
  - Immediately fails the tx if 0 on top of the stack

```
txna ApplicationArgs 0
btoi
assert
```

New Swap Opcode

- swap
  - Switches the top two items on the stack

```
// Teal Version 2 incrementing a global integer
byte "Total"
app_global_get
gtxn 1 Amount
+
store 1
byte "Total"
load 1
app_global_put

//Teal Version 3
byte "Total"
app_global_get
gtxn 1 Amount
+
byte "Total"
swap
app_global_put
```

New Dig N Opcode

- dig N
  - Make a copy of the Nth element in the stack and move to top

```
byte "first"
byte "second"
byte "third"
byte "fourth"
dig 3
```

GetBit and SetBit Opcodes

- setbit
- getbit
  - Maximize storage and allow setting and getting of individual bits

```
// check that the 6th bit is set in a unit64
int 235 // target
int 6 //bit to get
getbit
int 1
==
….
// global byte slices can store up to 64 bytes, set
// the 400th bit to 1
// this will panic if there is no 400th bit
int 0 //index into apps array
byte “myglobal64byteglobal”
app_global_get_ex
assert
int 400 //bit to  set
int  1 //value to set it to
setbit
```

GetByte and SetByte Opcodes

- setbyte
- getbyte
  - Allows getting and setting individual bytes

```
// Get the third byte in the following string
byte “Test the getbyte opcode” // target
int 3 // byte to retrieve
getbyte
int 116 //ASCII value for ‘t’
==
….
// Change the third byte in the following byte slice to ASCII ‘i’
byte “john” // target
int 2 // byte to set
int 105 // value to set it to “i”
setbyte
byte “join”
==
```

Select Opcode

- select
  - Conditional logic check. If top of stack is > 0 choose second element in the stack else third element in stack

```
int 1 // 0 value selection
byte “this is a select test” // not equal to 0 selection
int 1 // condition to check
select // finishes with the byte string on the top of the stack
```

Pushint and Pushbytes Opcodes

- pushint
- pushbytes
  - Performance increase
  - Mainly for tool builders

```
// load int 5 onto the top of the stack
pushint 5
// load byte slice onto the top of the stack
pushbytes “this is a byte slice”
```

Minimum Balance Opcode

- min_balance
  - Used to verify transaction(s) will not violate calculated minimum balance

```
// load application call senders balance
int 0
balance
// load a spend transaction from the same sender
// and subtract
gtxn 1 Amount
-
// load second transaction fee and subtract
gtxn 1 Fee
-
// load first transaction free and subtract
txn Fee
-
// get senders min balance
int 0
min_balance
// verify result of subtractions is greater than or equal to min balance
>=
```

Demo

Relative Transactions

Relative Group Transaction Indirection

- gtxns
- gtxnsa
  - Allow calculated transaction checking
  - Enhances gtxn and gtxna

```
//TEAL Version 2
gtxn 1 Amount // hard coded transaction number
int 10000
>

//TEAL Version 3
txn GroupIndex // Current group index
int 1
+
gtxns Amount

….

// get the first argument of the first transaction
// in a stateful smart contract
int 0
gtxnsa ApplicationArgs 0
```

Demo

Follow Guidelines for Safety

Boilerplate Approval Program

Example Stateless Contract

Adhere to These Guidelines to Protect Accounts

Size Limitations and Opcode Cost Limits

Presentation and Examples

https://github.com/algorand/smart-contracts/tree/master/devrel/teal3

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