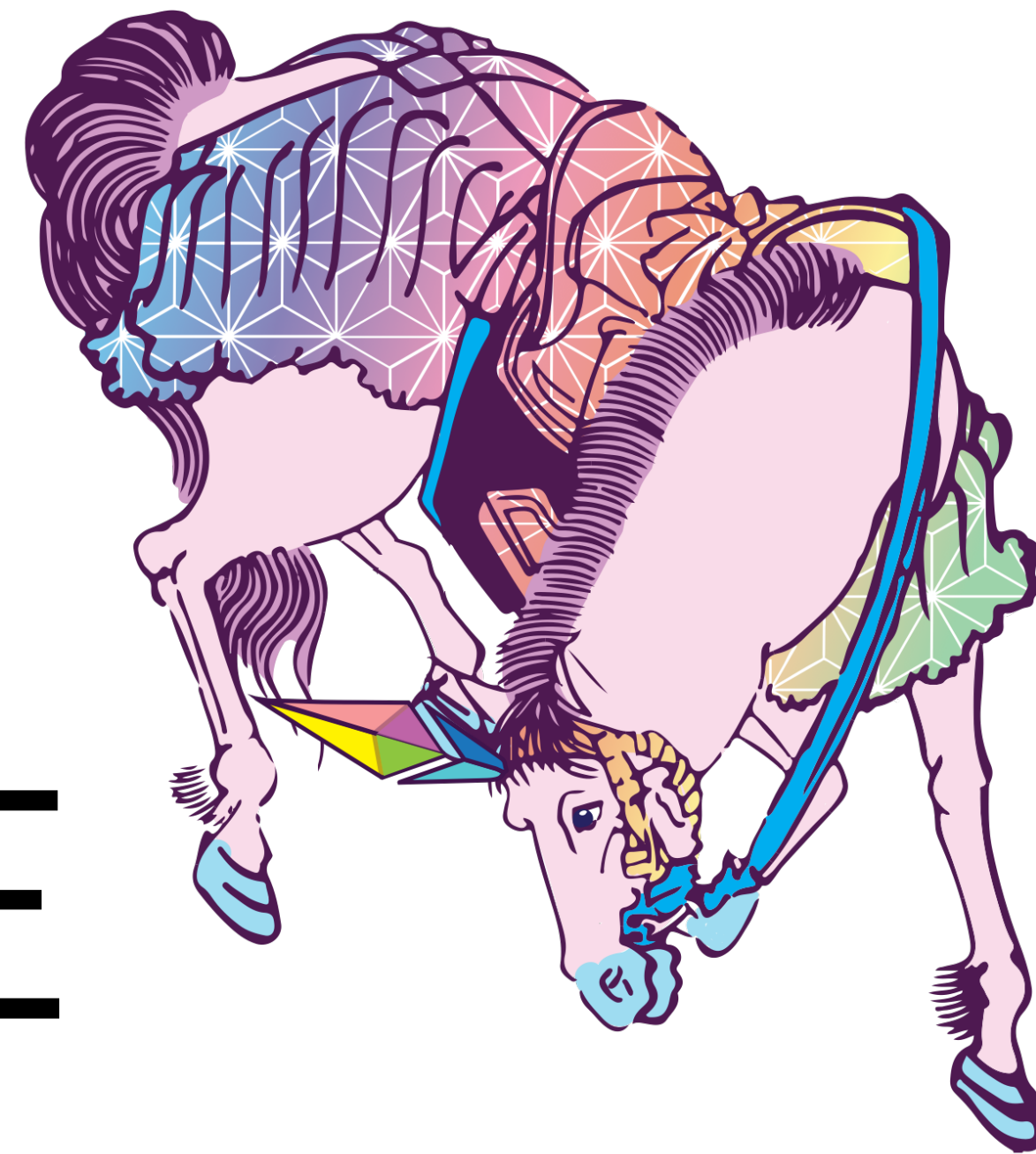
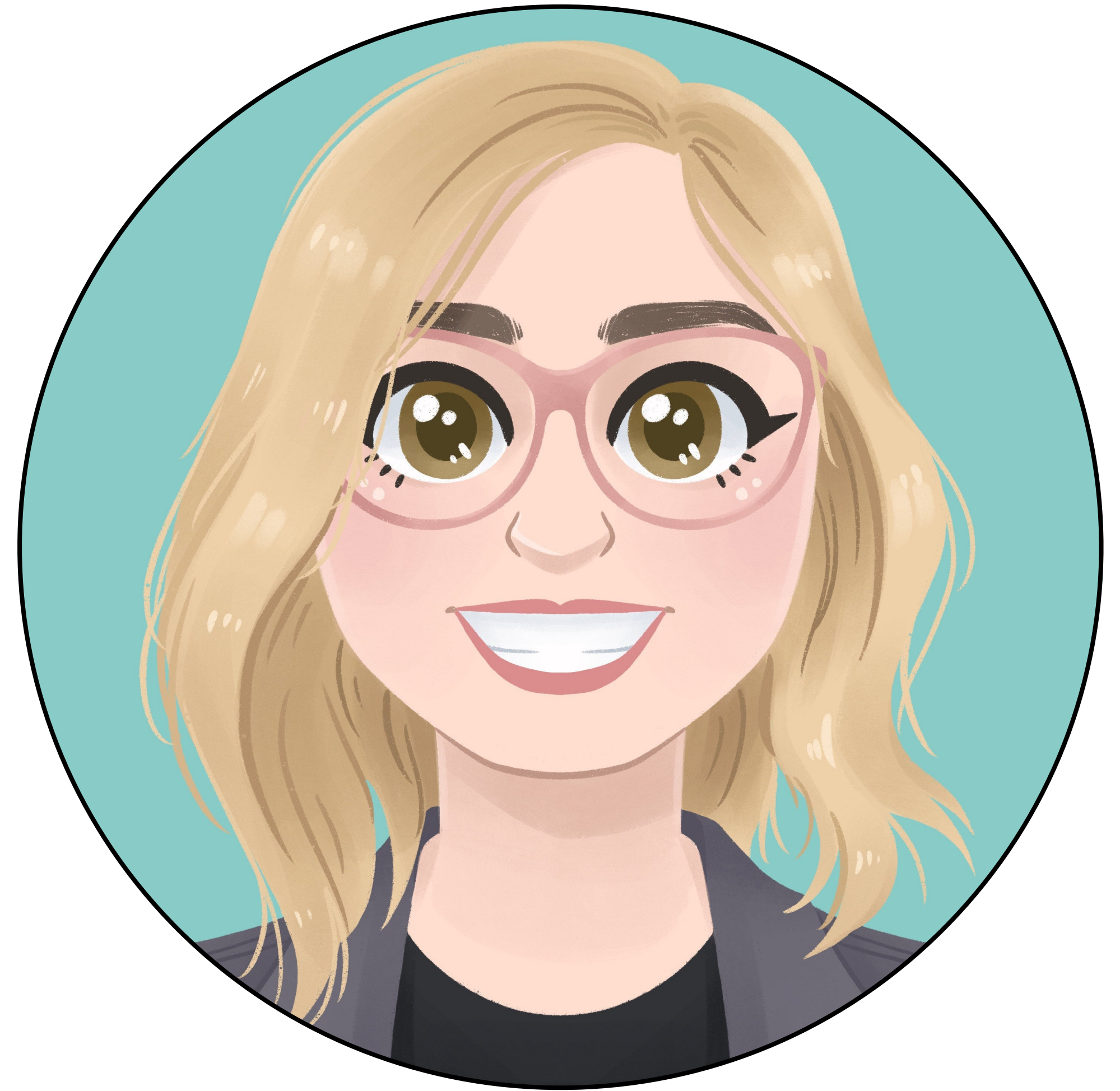


SMART CONTRACTS AS A SERVICE



 *USING MESSAGES TO SHARE CODE*

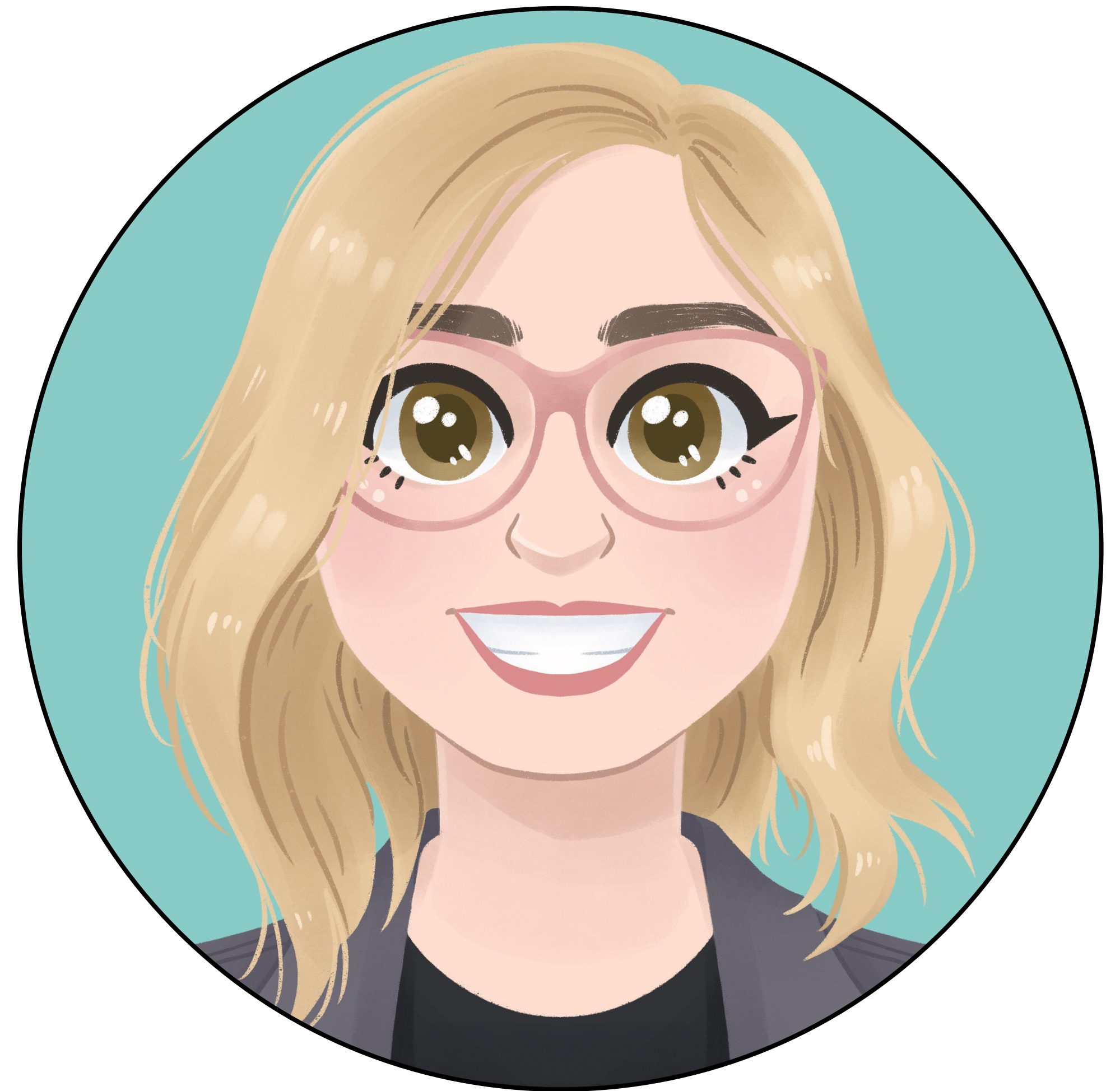
SMART CONTRACTS AS A SERVICE
BROOKLYN ZELENKA, @expede



SMART CONTRACTS AS A SERVICE

BROOKLYN ZELENKA, @expede

- Cofounder/CTO at Fission — <https://fission.codes>
- It is **so incredibly good** to be back in Osaka 🎉
- PLT & VM enthusiast
- Prev. Ethereum R&D, now visiting from IPFS-land 🚀🌐
- EIPs
 - 615: *EVM Subroutines & Static Jumps*
 - 902: *Token Permissions / Validation*
 - 1066: *Standardized Status Codes*
 - *Dependency for security token standards (ERC-1400), &c*
 - 1444: *Permissionless On-Chain Translation*
- ECIP 1050





ETC Cooperative
@ETCCooperative



Very excited about the [@etclabs](#) + [@gitcoin](#) collaboration!

We plan to provide ETC bounties for:

1. EVM target versioning for Vyper so we can use Vyper for ETC ([@fubuloubu](#))
2. ERC-1066 native support in Vyper and Solidity (SOLC and SOLL) - [@ethchris](#)
3. ERC-1066 translations.



Ethereum Classic Labs [@etclabs](#) · 8h

At #Devcon5, ETC Labs announced partnerships with [@gitcoin](#), [@SWARM](#) & [@bloqinc](#) to expand the collaboration between #ETC & #ETH. This will strengthen both communities and ensure developers the infrastructure they need to deliver on the promise of blockchain. ow.ly/Hx9630pHbRN



ETC Cooperative
@ETCCooperative



Very excited about the [@etclabs](#) + [@gitcoin](#) collaboration!

We plan to provide ETC bounties for:

1. EVM target versioning for Vyper so we can use Vyper for ETC ([@fubuloubu](#))
2. **ERC-1066 native support in Vyper and Solidity** (SOLC and SOLL) - [@ethchris](#)
3. **ERC-1066 translations.**



Ethereum Classic Labs [@etclabs](#) · 8h

At #Devcon5, ETC Labs announced partnerships with [@gitcoin](#), [@SWARM](#) & [@bloqinc](#) to expand the collaboration between #ETC & #ETH. This will strengthen both communities and ensure developers the infrastructure they need to deliver on the promise of blockchain. ow.ly/Hx9630pHbRN

WEB OF CONTRACTS

WEB OF CONTRACTS



WEB OF CONTRACTS

BIG, HAIRY, AUDACIOUS GOAL

WEB OF CONTRACTS

BIG, HAIRY, AUDACIOUS GOAL

- Write fewer lines of code 🕒

WEB OF CONTRACTS

BIG, HAIRY, AUDACIOUS GOAL

- Write fewer lines of code 🕒
- Have higher confidence 🚀

WEB OF CONTRACTS

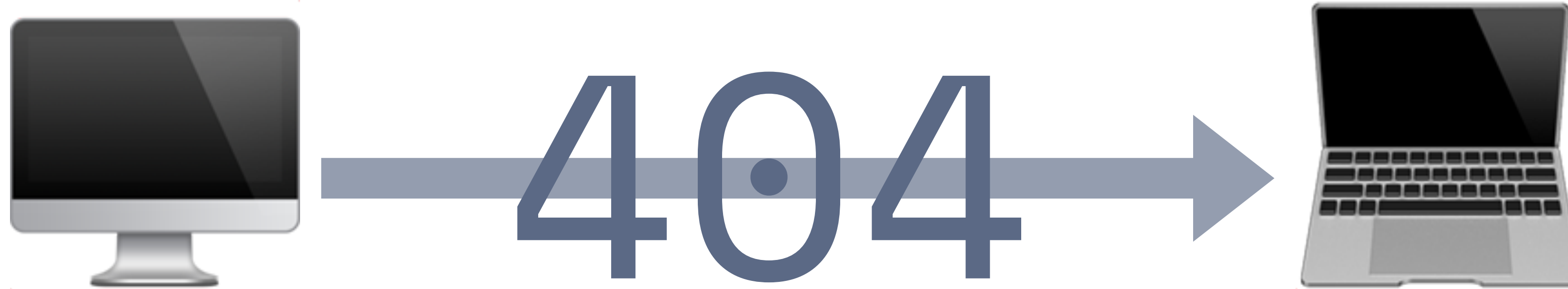
BIG, HAIRY, AUDACIOUS GOAL

- Write fewer lines of code 🕒
- Have higher confidence 🚀
- Make Ethereum more accessible 👩👧👦

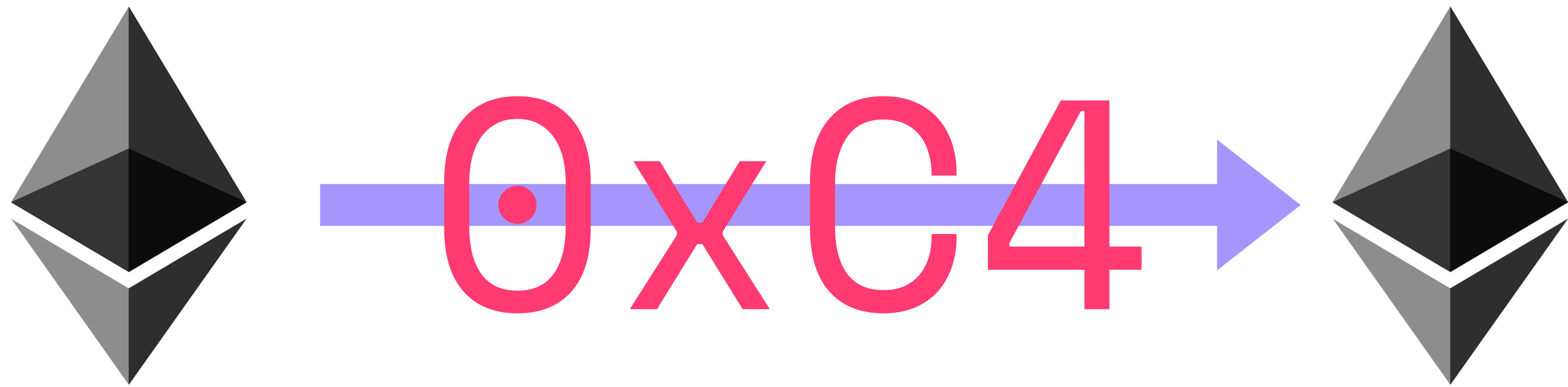
WEB OF CONTRACTS

HTTP → ETH

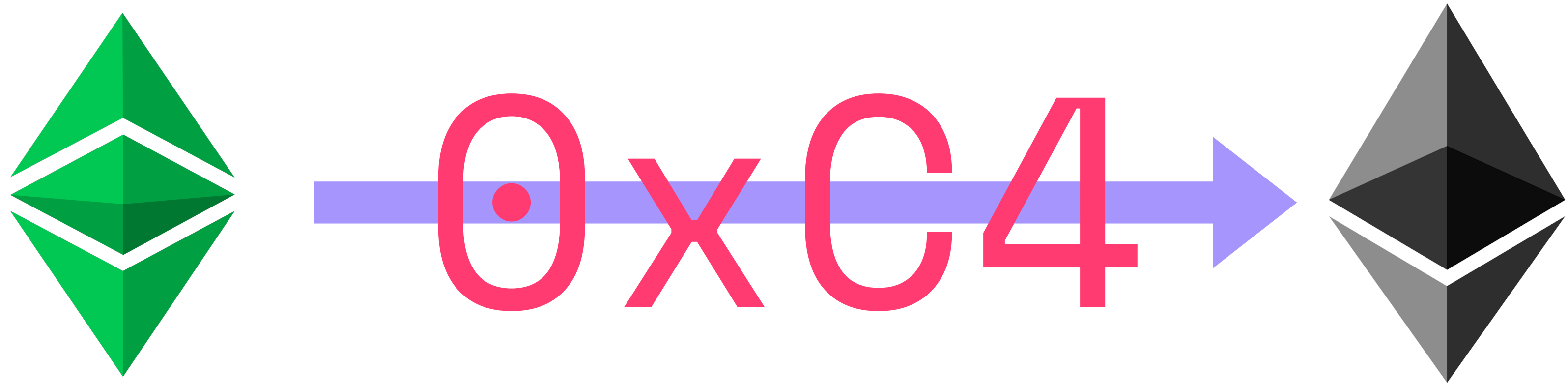
WEB OF CONTRACTS
HTTP → ETH



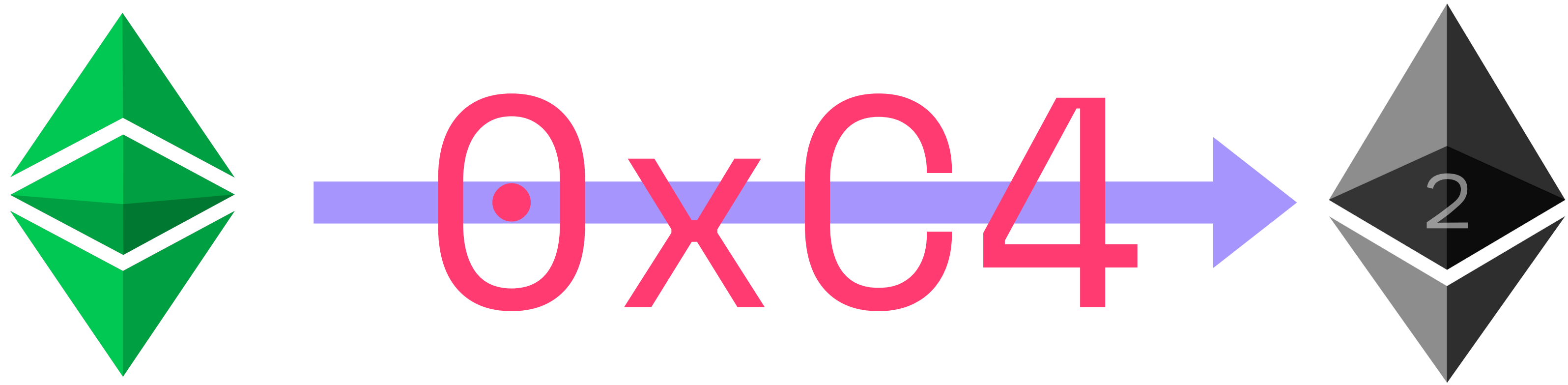
WEB OF CONTRACTS
HTTP → ETH



WEB OF CONTRACTS
HTTP → ETH



WEB OF CONTRACTS
HTTP → ETH



WEB OF CONTRACTS

A FUTURE ALONGSIDE UNIX & HTTP 🚀

WEB OF CONTRACTS

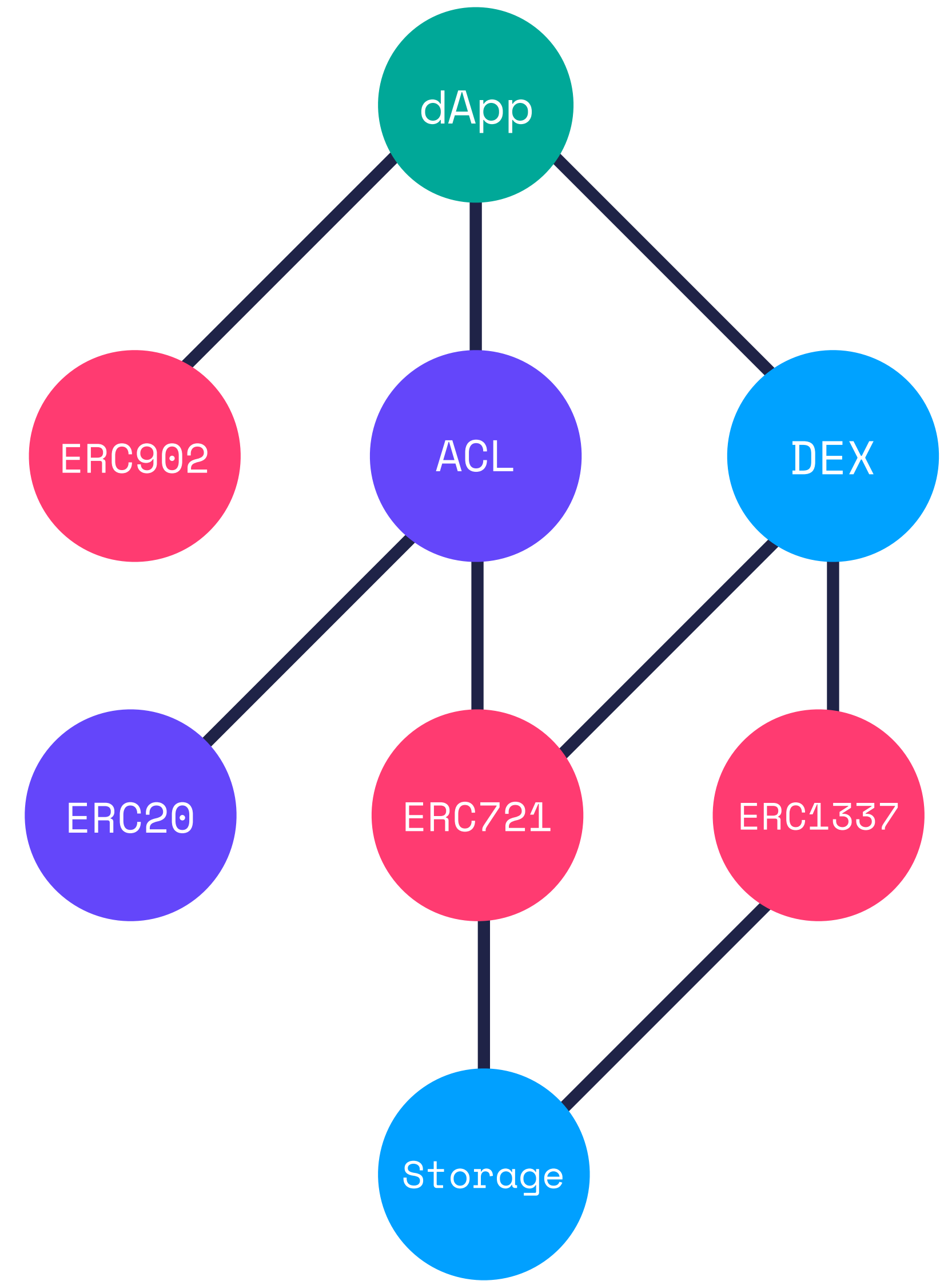
A FUTURE ALONGSIDE UNIX & HTTP 🚀

- What do Unix and HTTP have in common? **Composition!**
- Ethereum is a shared system — can we **leverage** each other's contacts?
- Can high value data & utilities **make money** on-chain?

WEB OF CONTRACTS

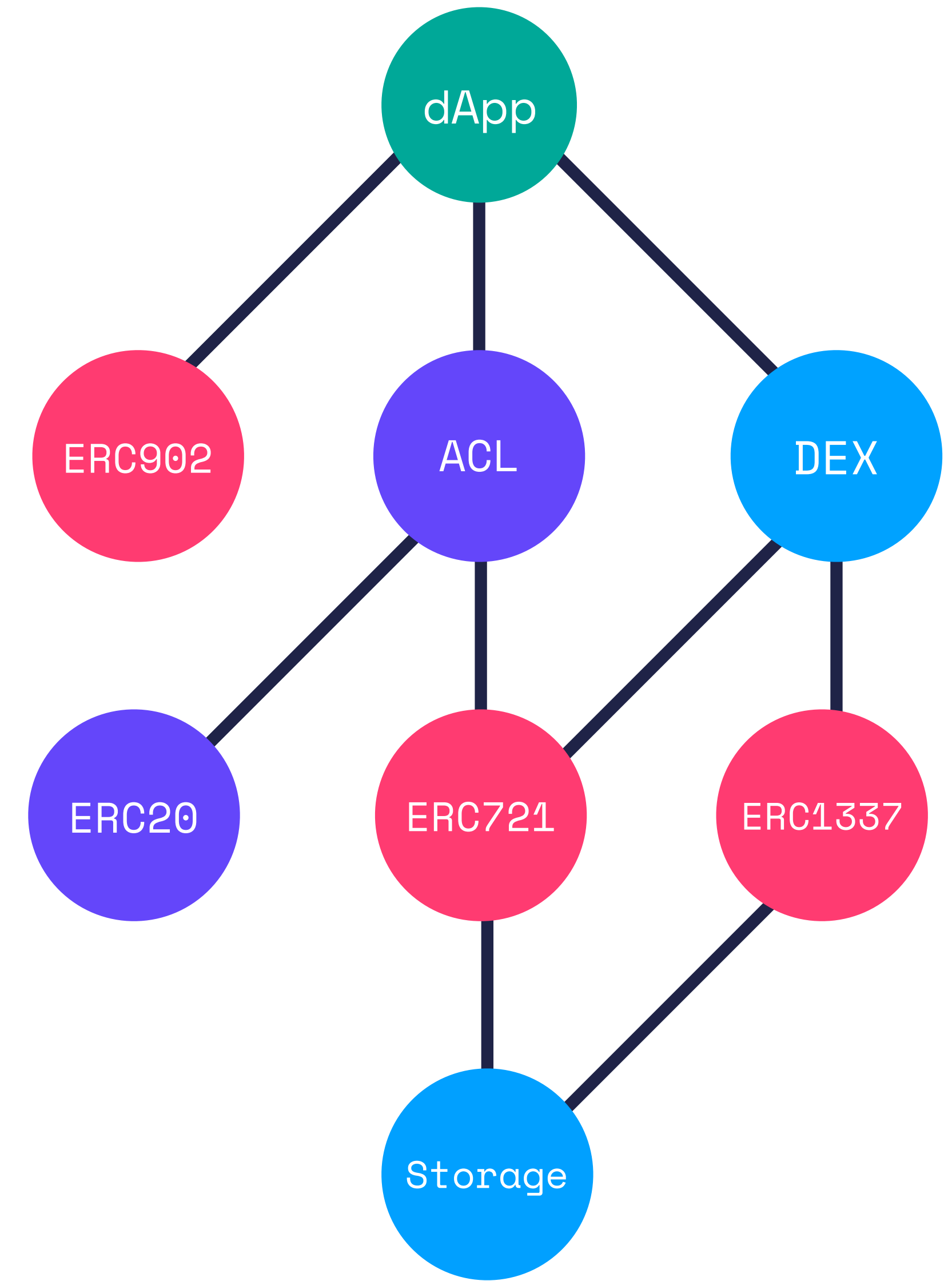
ON-CHAIN MICROSERVICES

WEB OF CONTRACTS
ON-CHAIN MICROSERVICES



WEB OF CONTRACTS ON-CHAIN MICROSERVICES

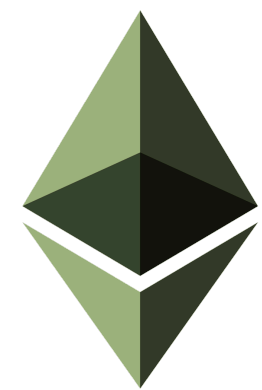
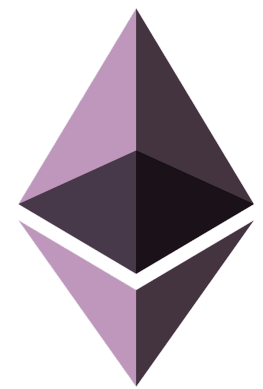
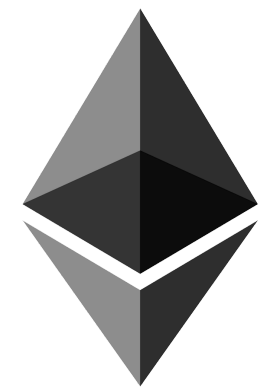
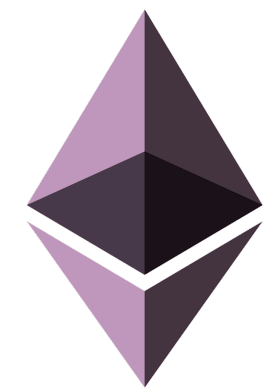
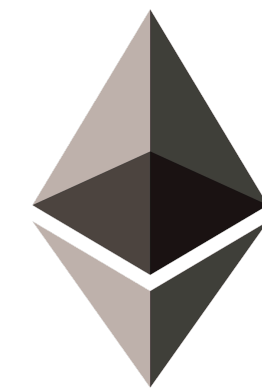
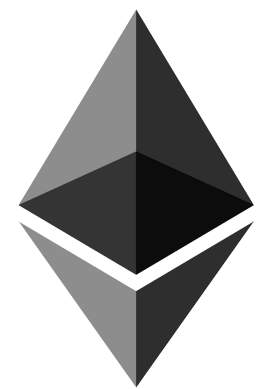
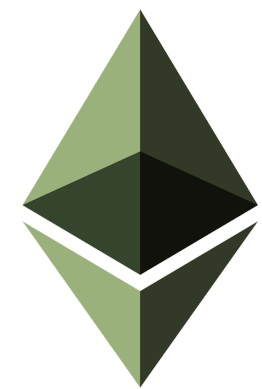
- “Web of contracts”
- Leverage shared infrastructure (pluggable)
- High quality smart contracts
- Efficient, reviewed, security hardened
- Compose new protocols
- Biz models for high-value service contracts



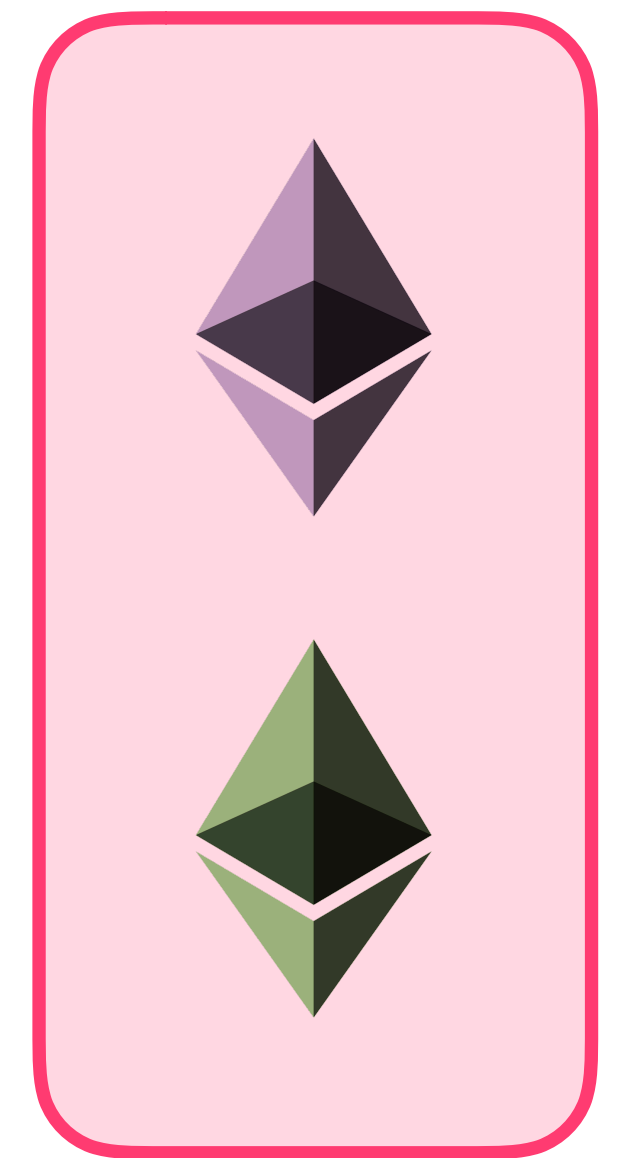
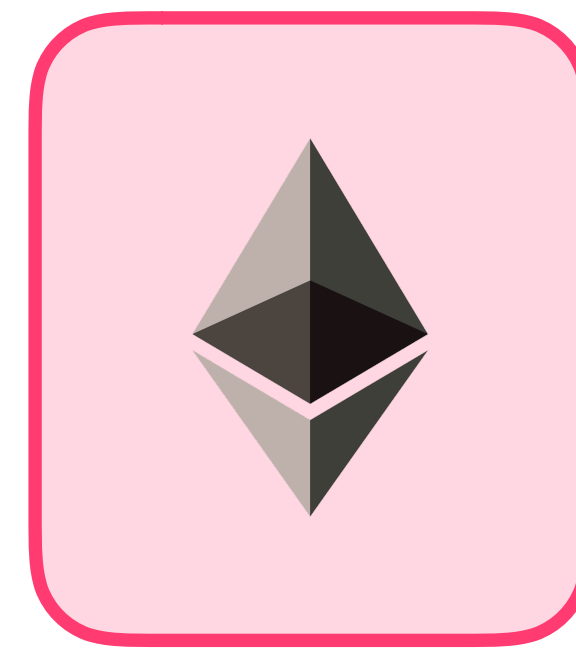
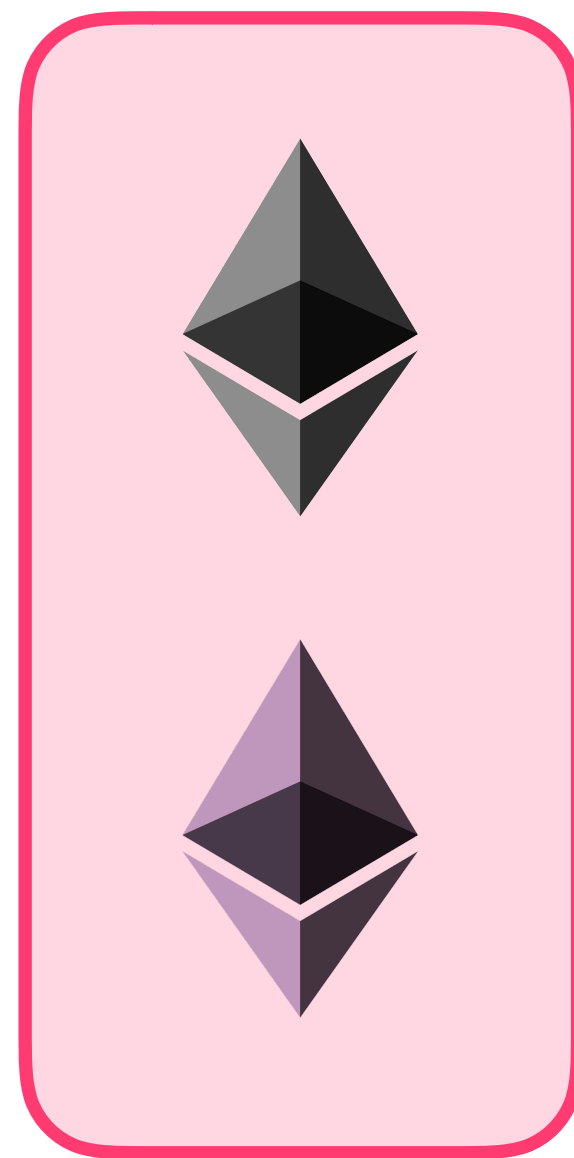
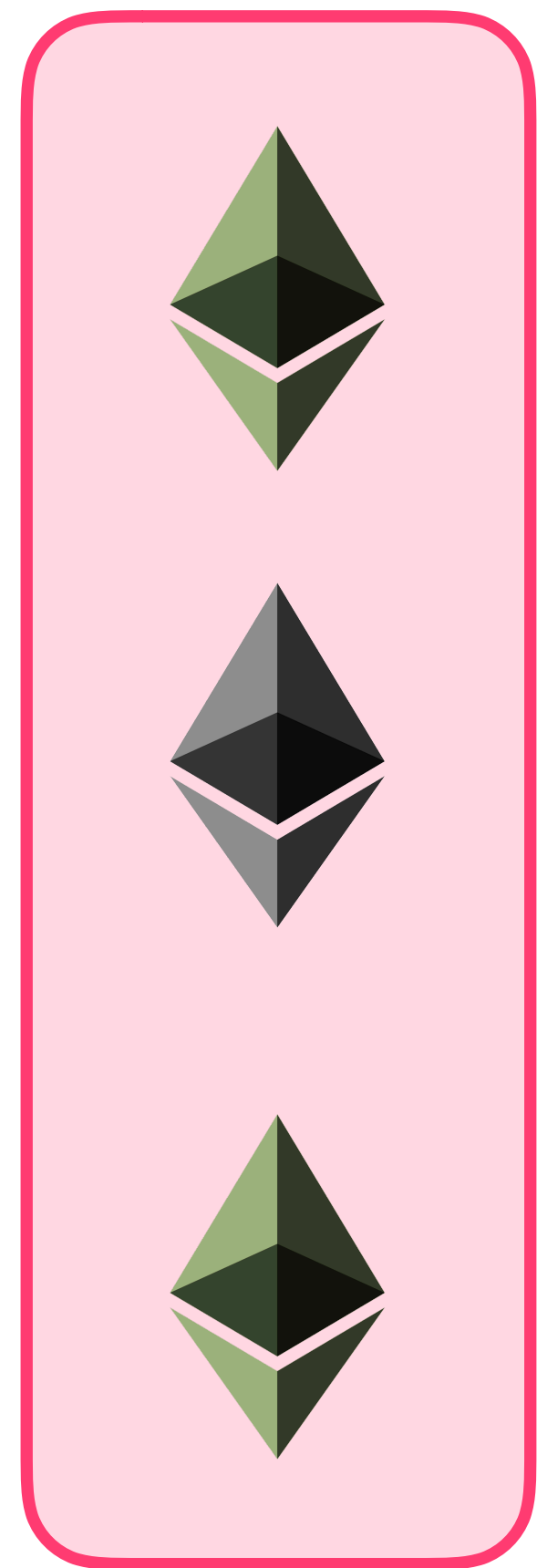
WEB OF CONTRACTS

THE CORE TECHNICAL IDEA

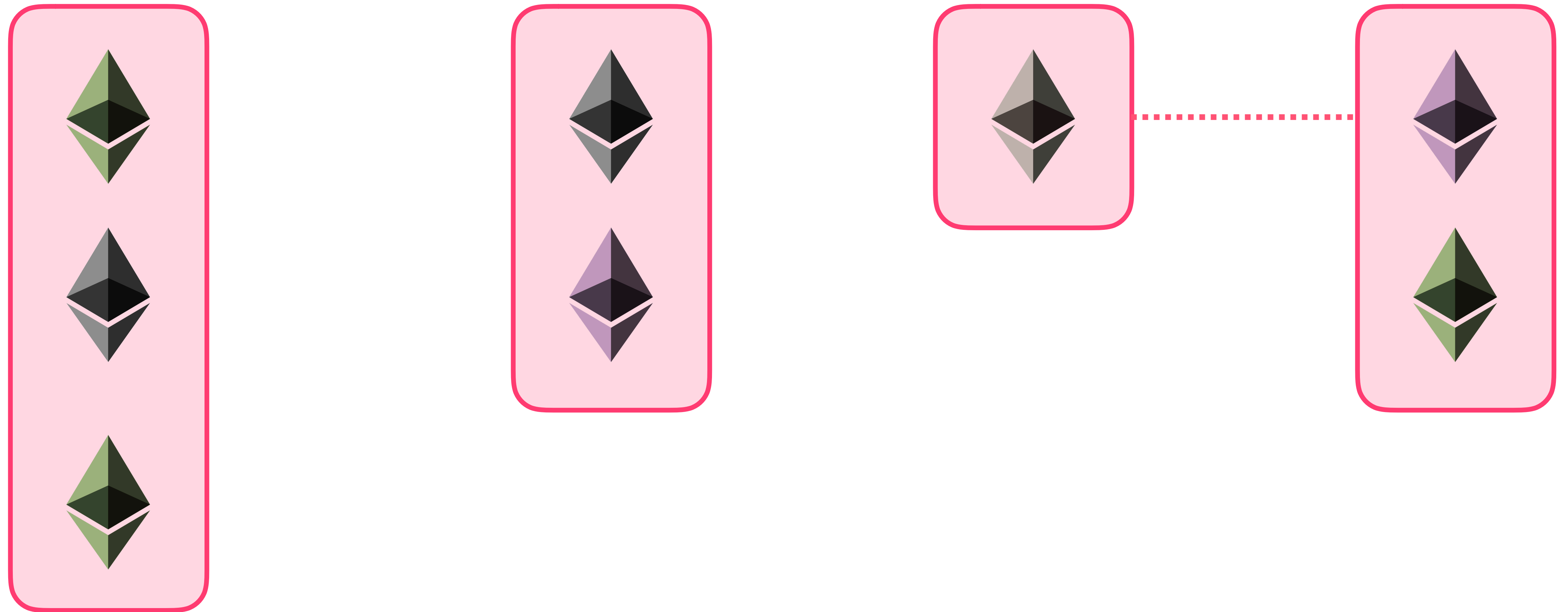
WEB OF CONTRACTS
THE CORE TECHNICAL IDEA



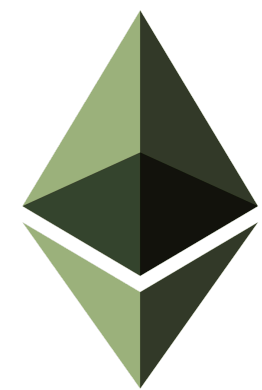
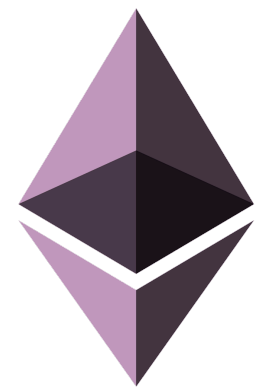
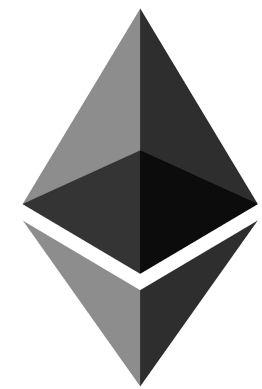
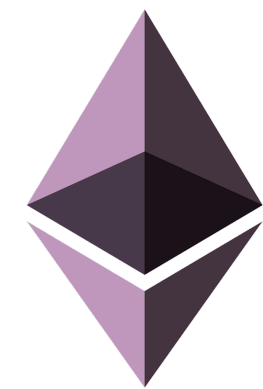
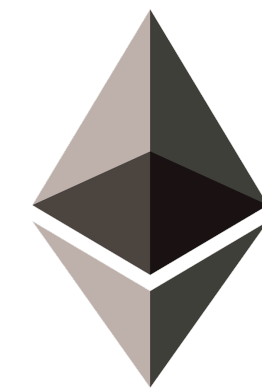
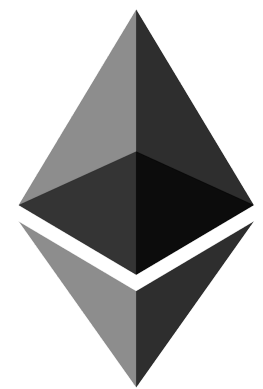
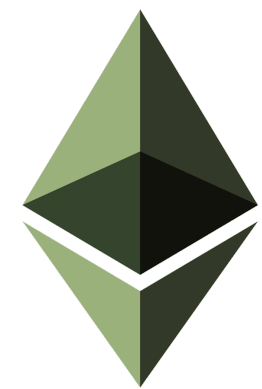
WEB OF CONTRACTS
THE CORE TECHNICAL IDEA



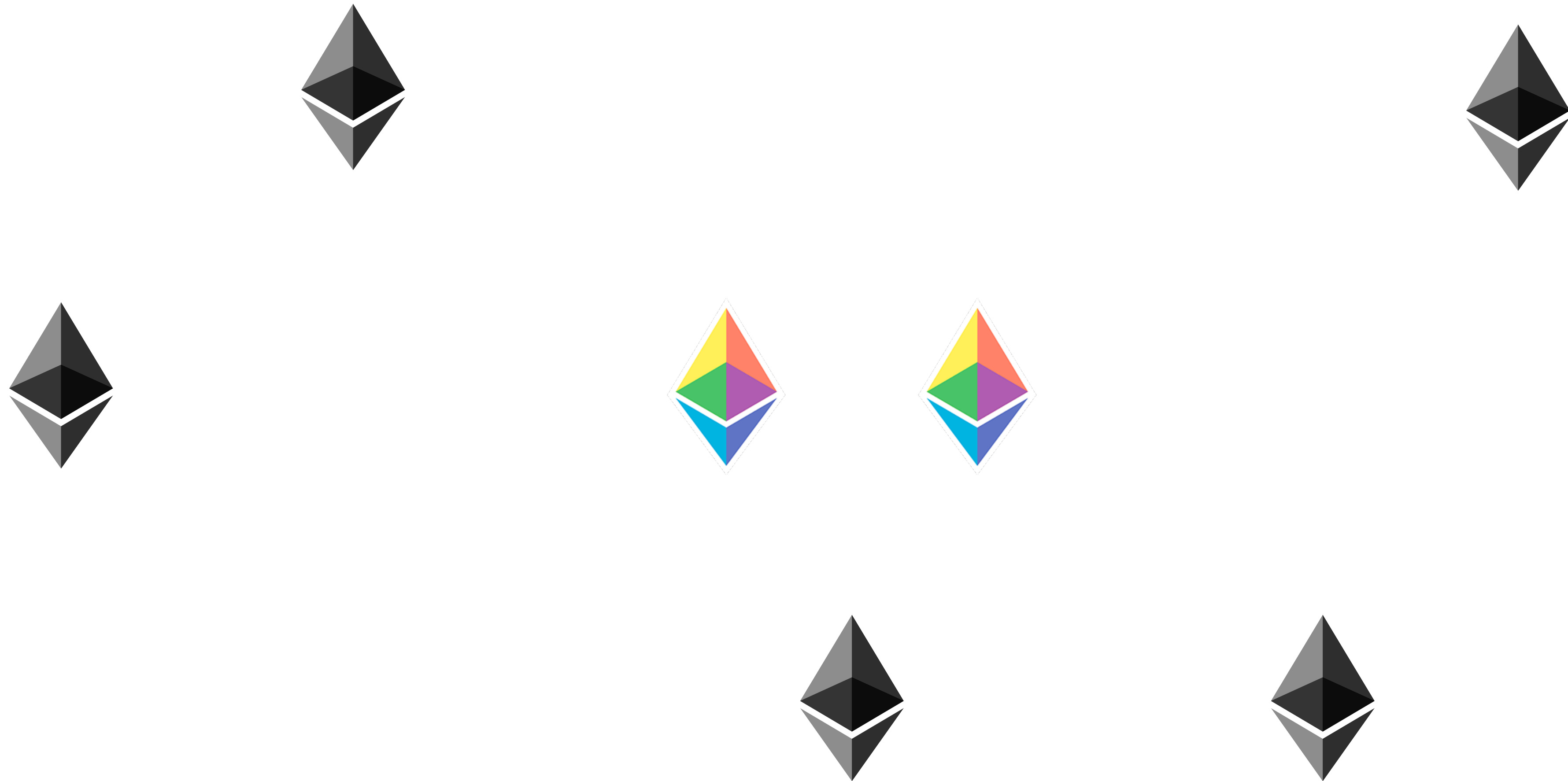
WEB OF CONTRACTS
THE CORE TECHNICAL IDEA



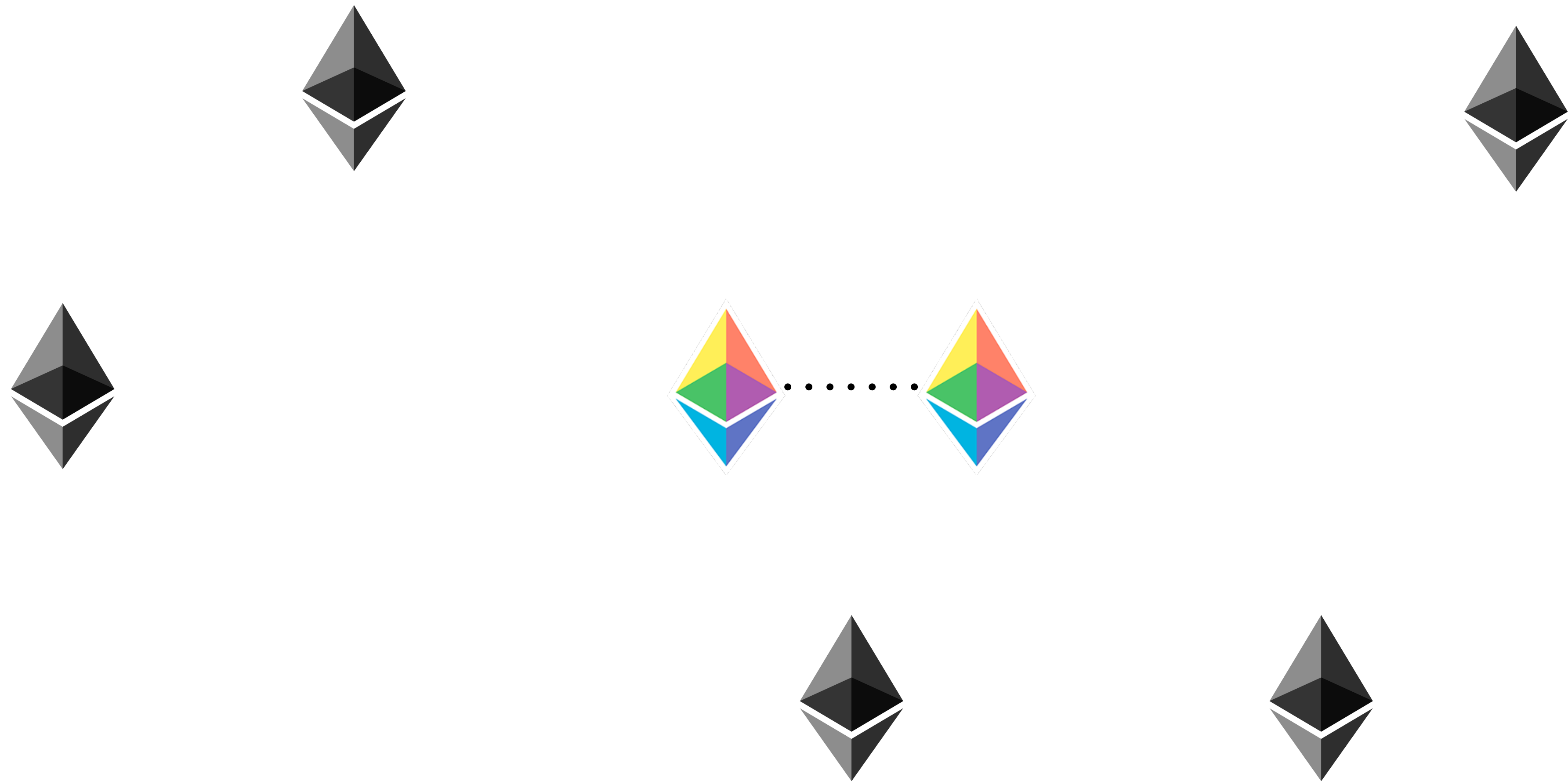
WEB OF CONTRACTS
THE CORE TECHNICAL IDEA



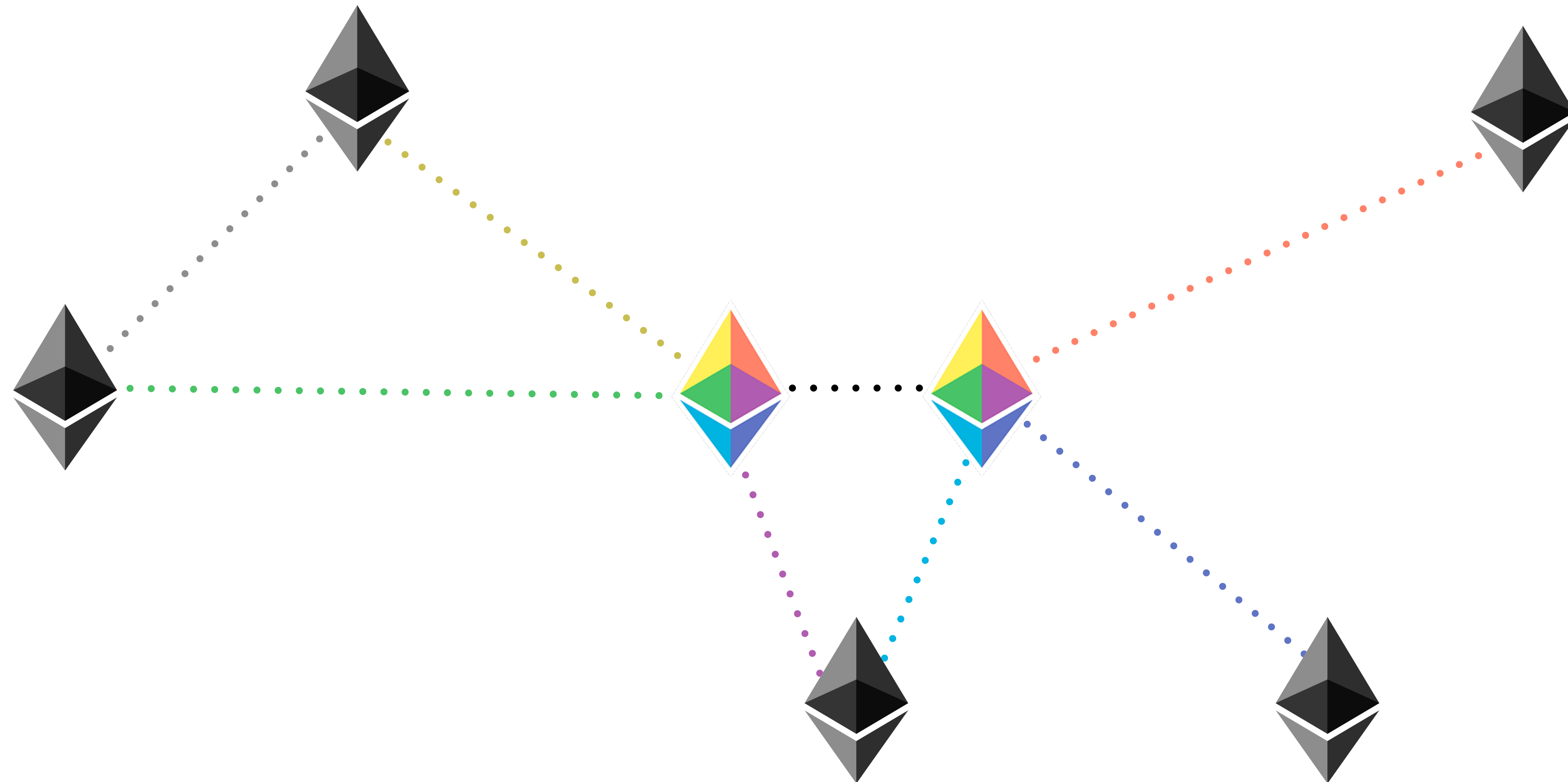
WEB OF CONTRACTS
THE CORE TECHNICAL IDEA



WEB OF CONTRACTS
THE CORE TECHNICAL IDEA



WEB OF CONTRACTS
THE CORE TECHNICAL IDEA



STRUCTURED STATUS CODES

STRUCTURED STATUS CODES

SO SIMPLE A MACHINE CAN USE IT 🤖👍

(HUMANS, TOO 😊👍)

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

OX41

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

0100 0001

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

Category 0100 0001

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

Category 0100 0001 Reason

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

Category 0100 0001 Reason

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

Category 0100 0001 Reason

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

Category 0100 0001 Reason

0X0B

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

Category 0100 0001 Reason

0X0B

0000 1011

STRUCTURED STATUS CODES

TASTY NIBBLES 🐭🧀

0X41

Category 0100 0001 Reason

0X0B

1011

(0xo* General is same as just the reason alone)

STRUCTURED STATUS CODES
CODE TABLE

STRUCTURED STATUS CODES CODE TABLE

Accepted/Started
Action Required

Fail
Ok
...

Reason

Category

	Time	Range	Auth	Agreements	Off Chain	...										
	0*	1*	2*	3*	4*	5*	6*	7*	8*	9*	A*	B*	C*	D*	E*	F*
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

SEARCH / MATCH

STRUCTURED STATUS CODES
CODE TABLE

Category

Time Range Auth Agreements Off Chain ...

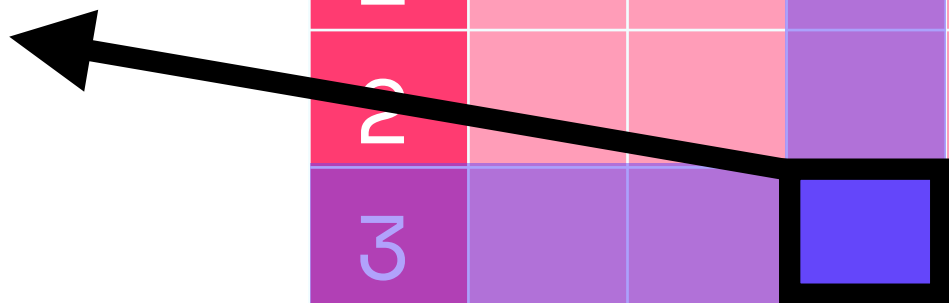
0x23 Awaiting Match

Fail
Ok
Accepted/Started
Action Required
...

Reason

	0*	1*	2*	3*	4*	5*	6*	7*	8*	9*	A*	B*	C*	D*	E*	F*
0																
1																
2																
3				AWAITING												
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

SEARCH / MATCH



STRUCTURED STATUS CODES

HELPER LIBRARY 

npmjs.com/package/fission-codes

STRUCTURED STATUS CODES

HELPER LIBRARY 

npmjs.com/package/fission-codes

```
enum Category {  
  Generic,  
  Permission,  
  Match,  
  Offer,  
  Availability,  
  // ...  
}
```

```
enum Reason {  
  Failure,  
  Success,  
  Acceptance,  
  Before,  
  ActionRequired,  
  // ...  
}
```

STRUCTURED STATUS CODES HELPER LIBRARY

npmjs.com/package/fission-codes

```
enum Category {  
  Generic,  
  Permission,  
  Match,  
  Offer,  
  Availability,  
  // ...  
}
```

```
enum Reason {  
  Failure,  
  Success,  
  Acceptance,  
  Before,  
  ActionRequired,  
  // ...  
}
```

```
function toCode(Category _category, Reason _reason) public pure returns (byte code) {  
  return toCode(uint(_category), uint(_reason));  
}  
  
function toCode(uint _category, uint _reason) public pure returns (byte code) {  
  return byte((_category << 4) + _reason);  
}  
  
function appCode(uint _appReason) public pure returns (byte code) {  
  return byte(160 + _appReason);  
}
```

Combine (inclusions)

STRUCTURED STATUS CODES HELPER LIBRARY

npmjs.com/package/fission-codes

```
enum Category {  
  Generic,  
  Permission,  
  Match,  
  Offer,  
  Availability,  
  // ...  
}
```

```
enum Reason {  
  Failure,  
  Success,  
  Acceptance,  
  Before,  
  ActionRequired,  
  // ...  
}
```

```
function toCode(Category _category, Reason _reason) public pure returns (byte code) {  
  return toCode(uint(_category), uint(_reason));  
}  
  
function toCode(uint _category, uint _reason) public pure returns (byte code) {  
  return byte((_category << 4) + _reason);  
}  
  
function appCode(uint _appReason) public pure returns (byte code) {  
  return byte(160 + _appReason);  
}
```

Combine (inclusions)

```
function categoryOf(byte _status) public pure returns (uint category) {  
  return uint(_status >> 4);  
}  
  
function reasonOf(byte _status) public pure returns (uint reason) {  
  return uint(_status & hex"0F");  
}
```

Pull Apart (projections)

STRUCTURED STATUS CODES HELPER LIBRARY

npmjs.com/package/fission-codes

```
enum Category {  
  Generic,  
  Permission,  
  Match,  
  Offer,  
  Availability,  
  // ...  
}
```

```
enum Reason {  
  Failure,  
  Success,  
  Acceptance,  
  Before,  
  ActionRequired,  
  // ...  
}
```

```
function toCode(Category _category, Reason _reason) public pure returns (byte code) {  
  return toCode(uint(_category), uint(_reason));  
}  
  
function toCode(uint _category, uint _reason) public pure returns (byte code) {  
  return byte((_category << 4) + _reason);  
}  
  
function appCode(uint _appReason) public pure returns (byte code) {  
  return byte(160 + _appReason);  
}
```

Combine (inclusions)

```
function categoryOf(byte _status) public pure returns (uint category) {  
  return uint(_status >> 4);  
}  
  
function reasonOf(byte _status) public pure returns (uint reason) {  
  return uint(_status & hex"0F");  
}
```

Pull Apart (projections)

```
function requireOk(byte _status, string message) public pure {  
  require(isOk(_status), message);  
}
```

Automatic **require**
with (hardcoded) message*

TOY FLOW FOR
CONTRACT MESSAGING

TOY FLOW FOR CONTRACT MESSAGING

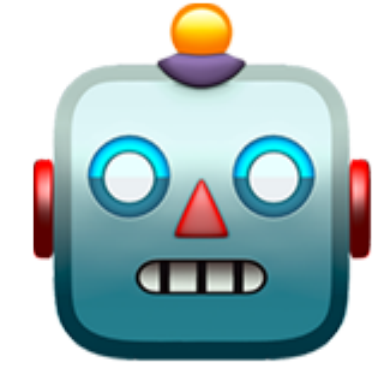


TOY FLOW

NOT FOR SALE... YET

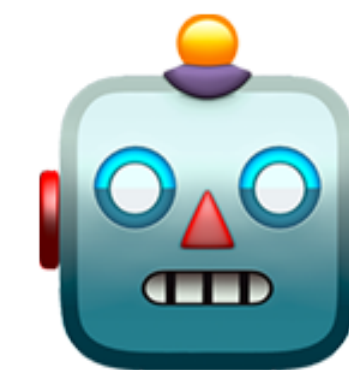
TOY FLOW

NOT FOR SALE... YET



TOY FLOW

NOT FOR SALE... YET

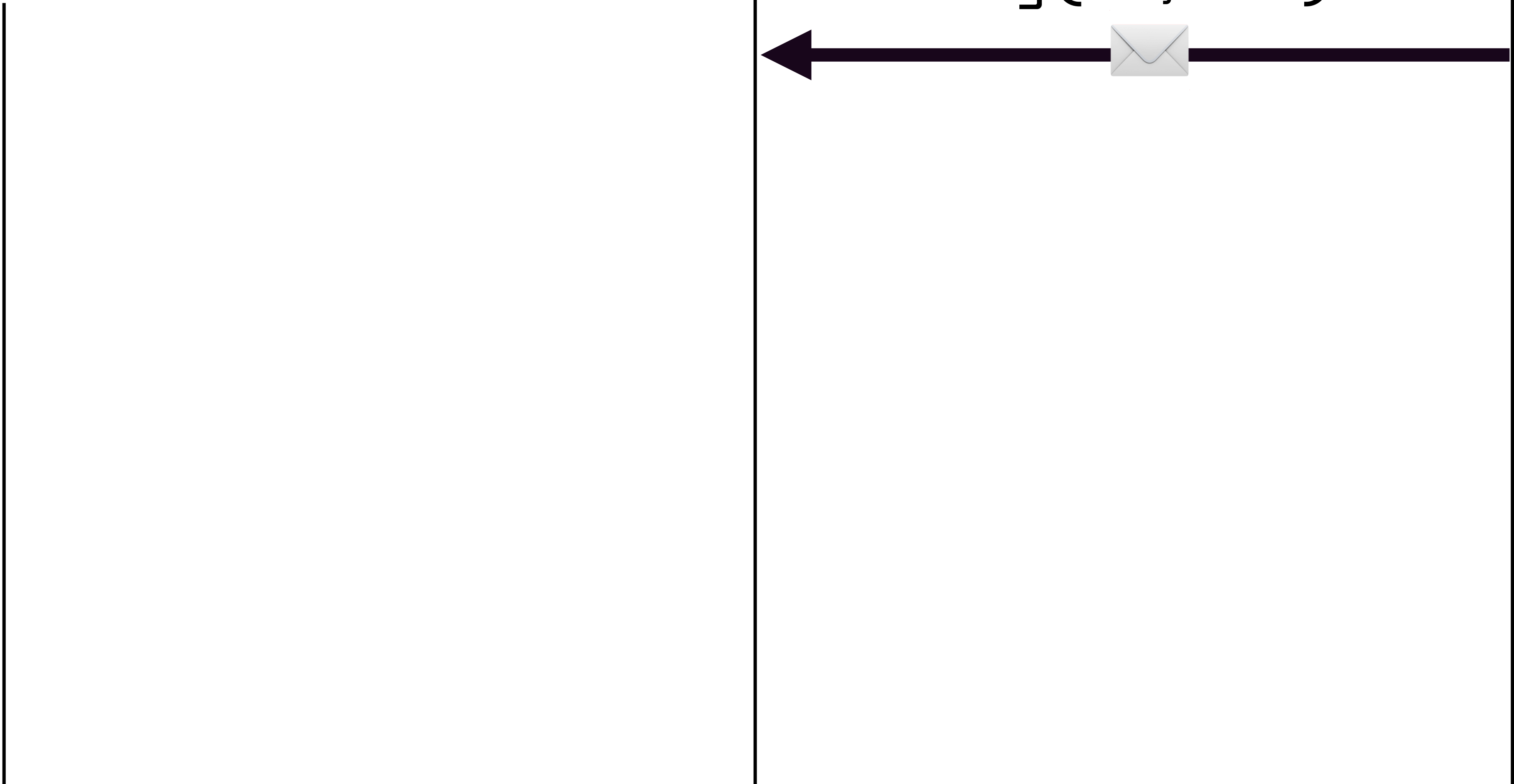
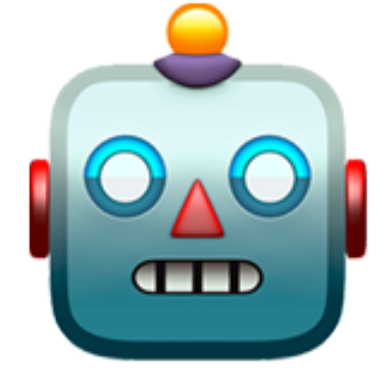


TOY FLOW

NOT FOR SALE... YET

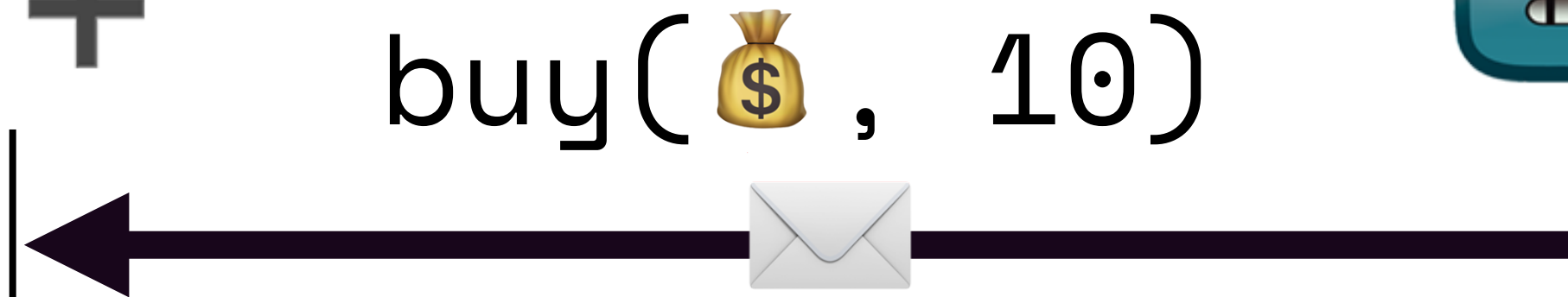
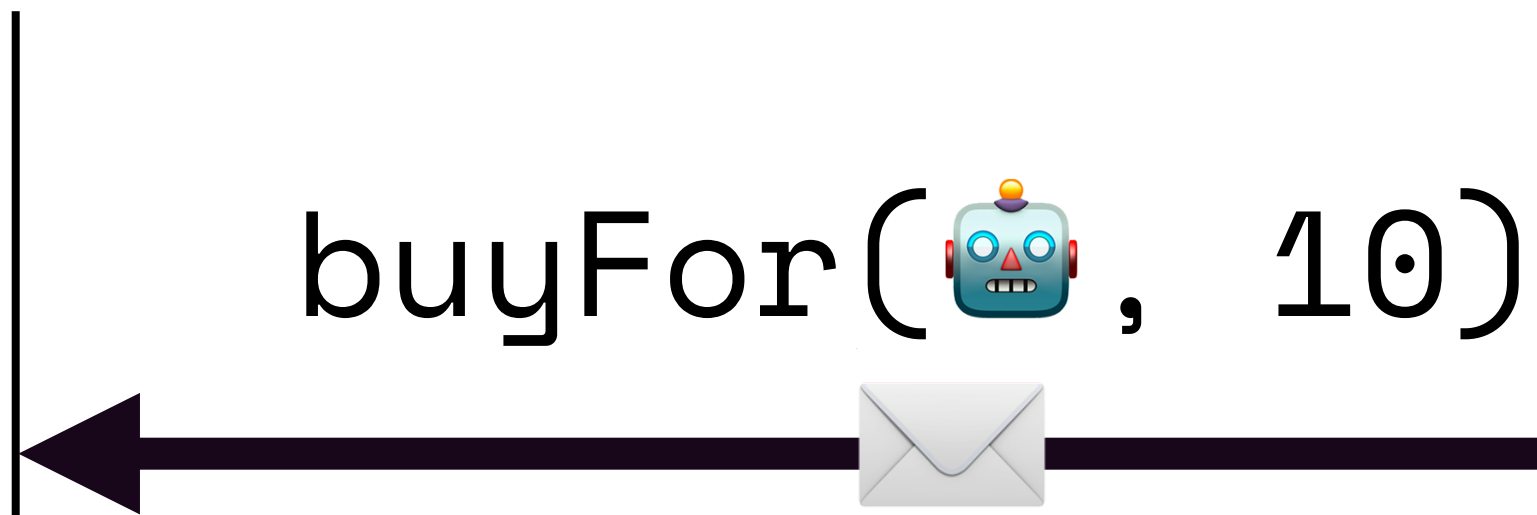
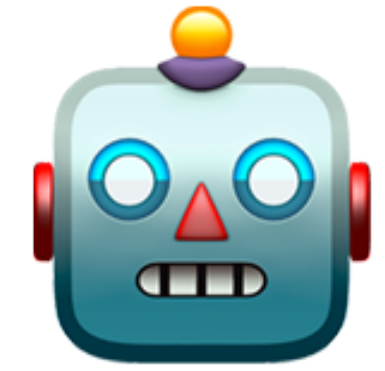


buy(💰, 10)



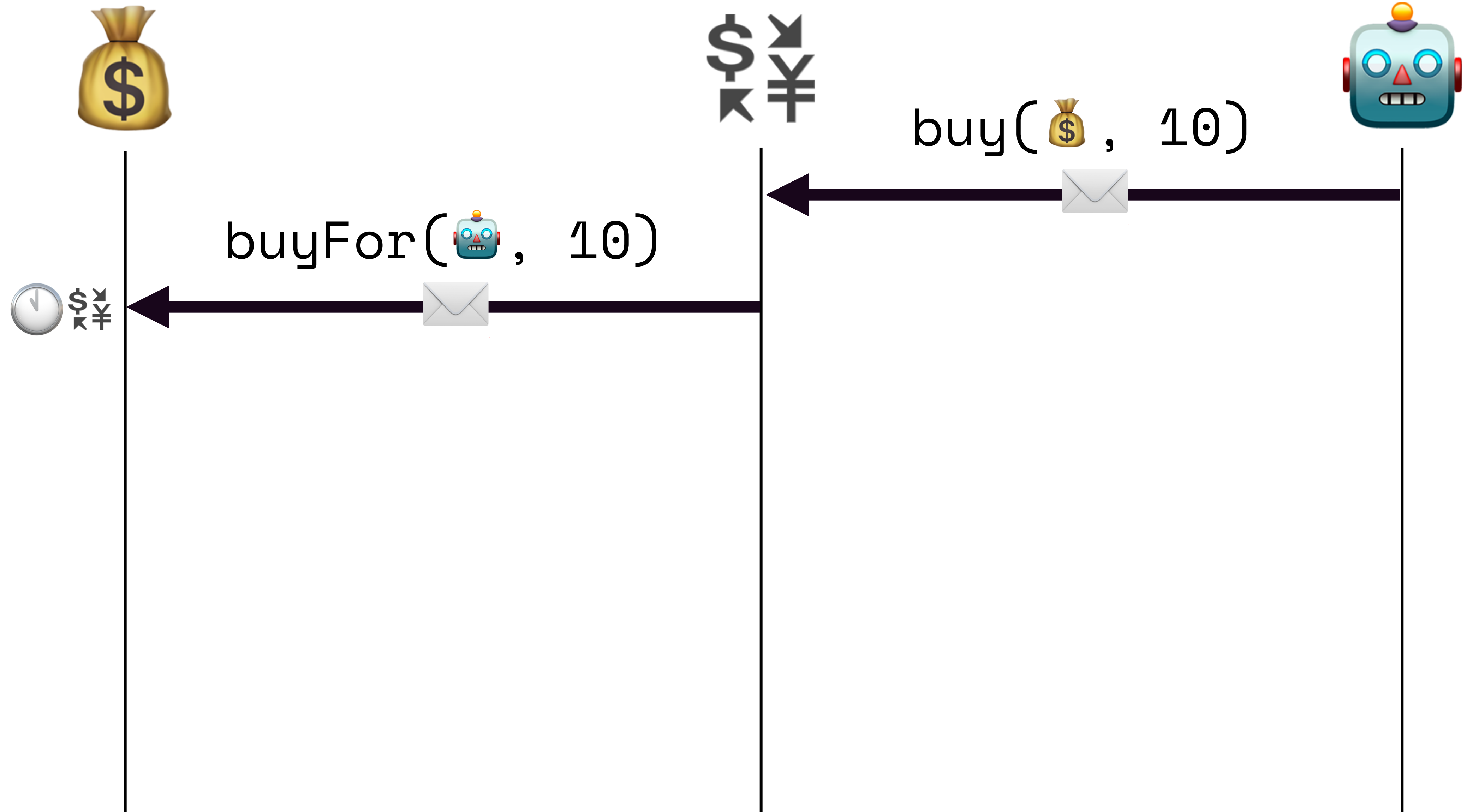
TOY FLOW

NOT FOR SALE... YET



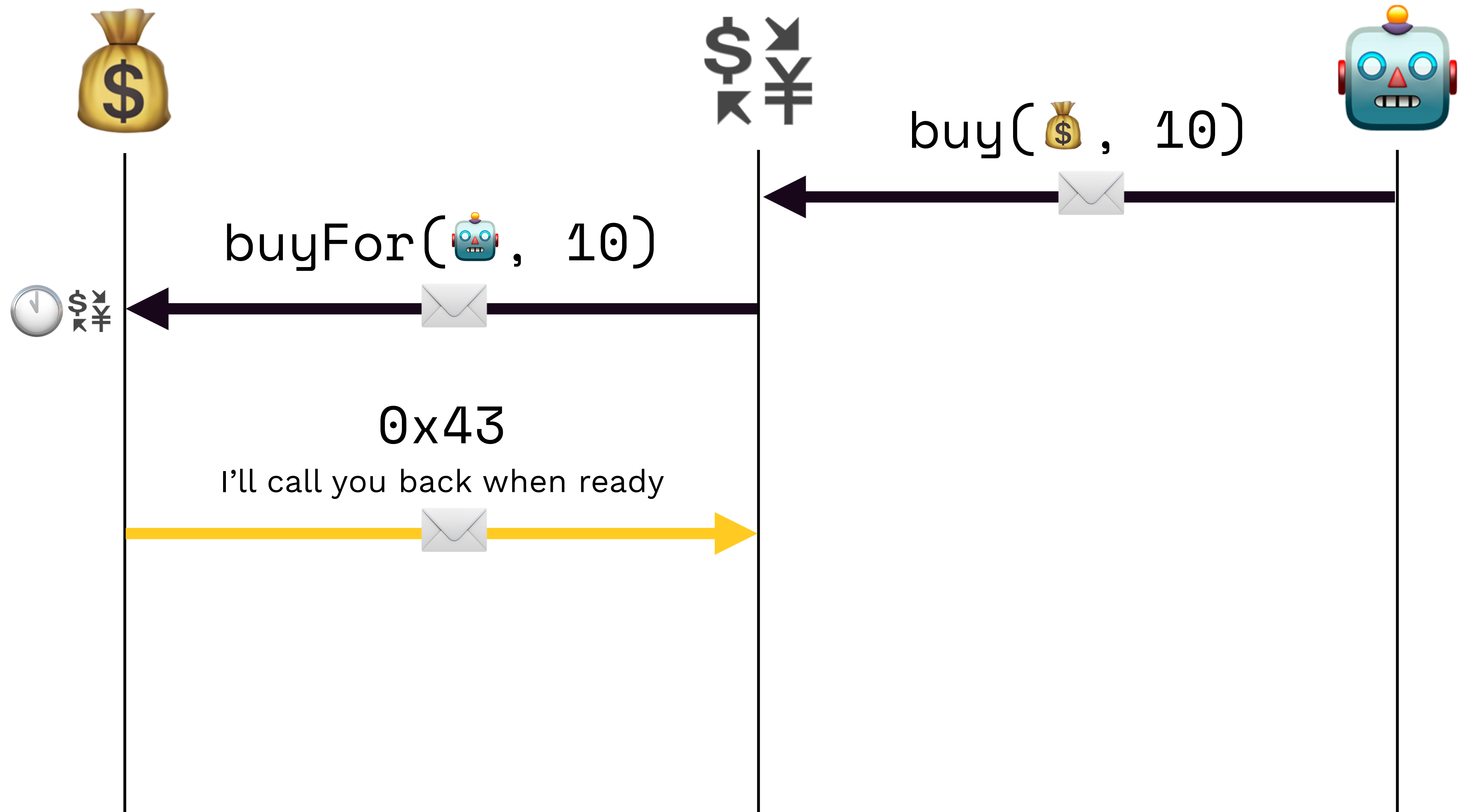
TOY FLOW

NOT FOR SALE... YET



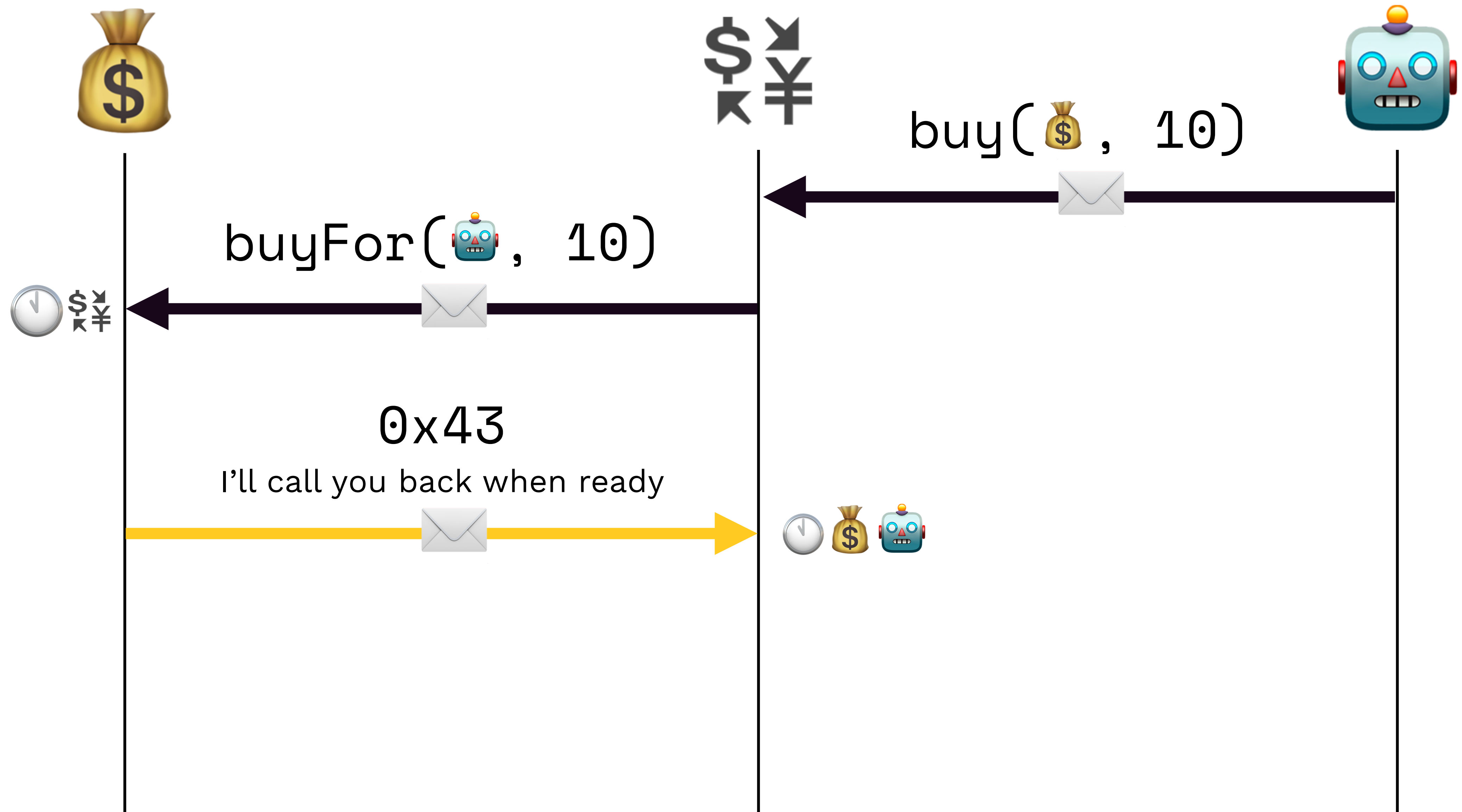
TOY FLOW

NOT FOR SALE... YET



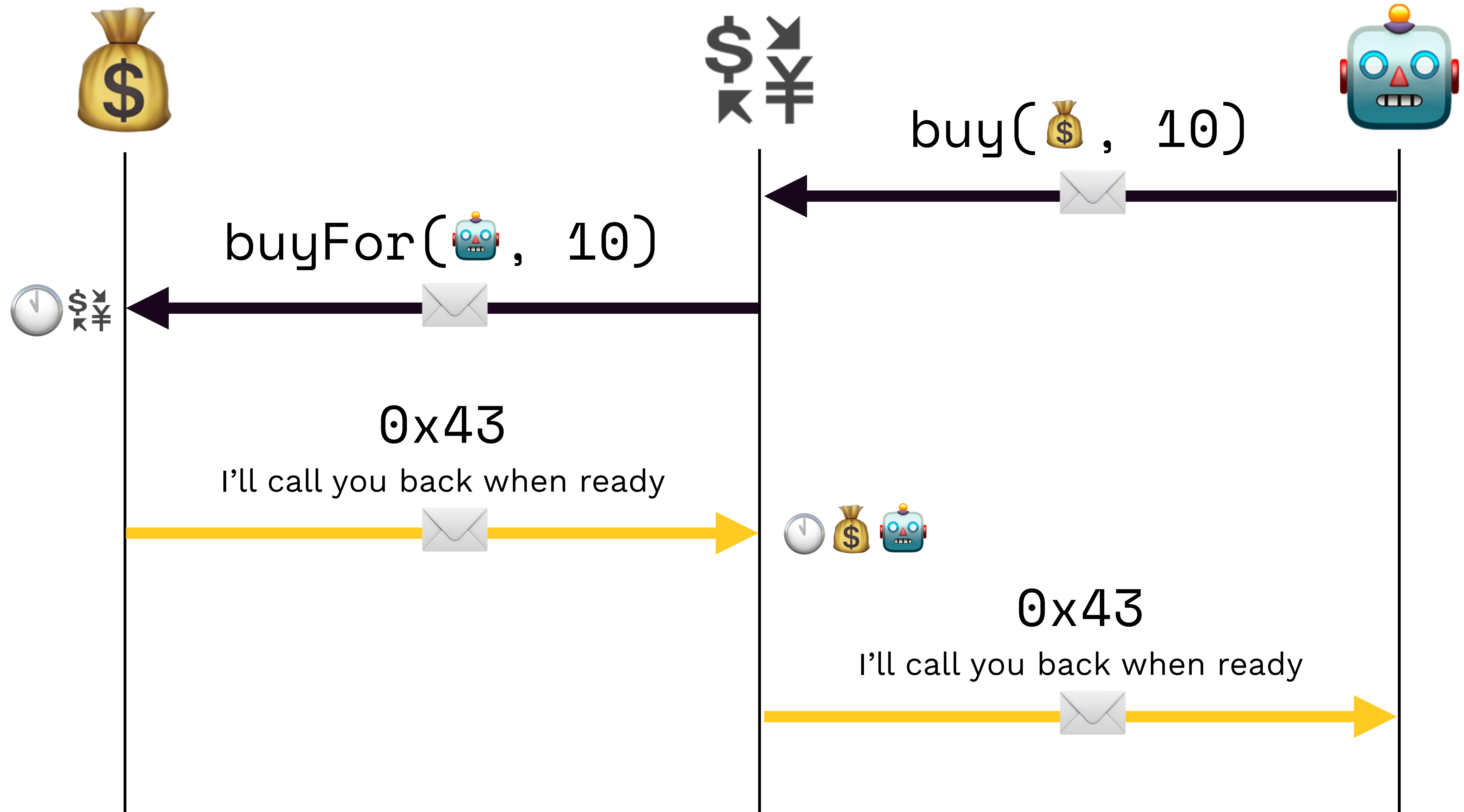
TOY FLOW

NOT FOR SALE... YET

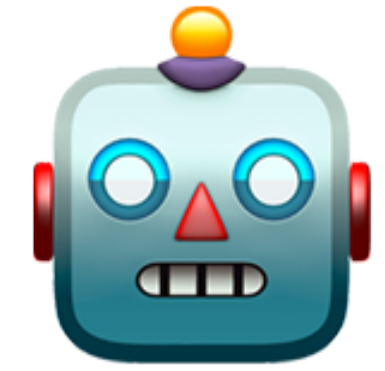


TOY FLOW

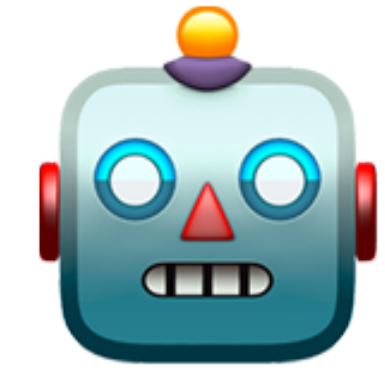
NOT FOR SALE... YET



TOY FLOW IMPATIENCE



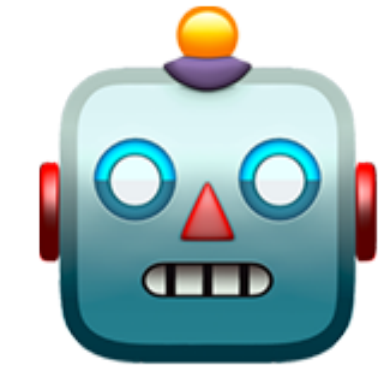
TOY FLOW IMPATIENCE



isDoneYet



TOY FLOW IMPATIENCE



isDoneYet 

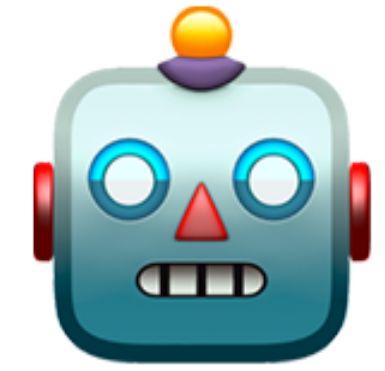
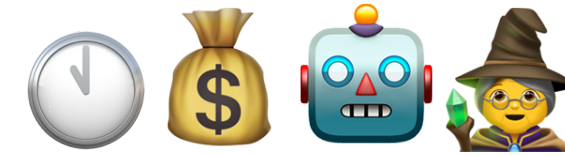


0x43

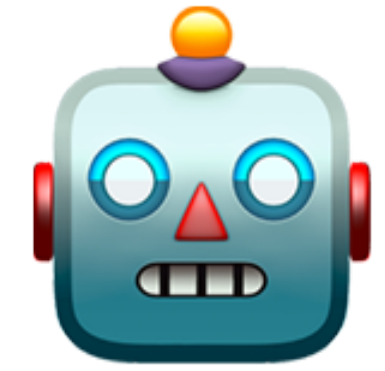
I'll call you back when ready



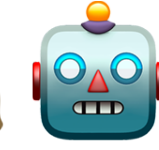
TOY FLOW
KICK-OFF



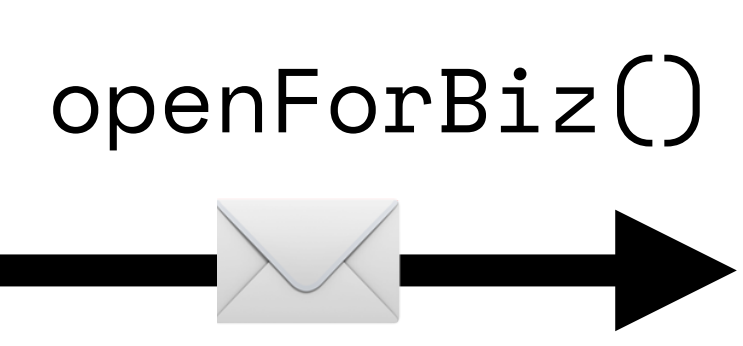
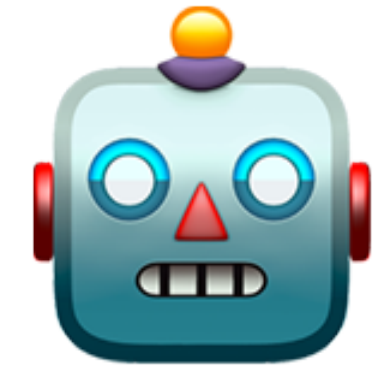
TOY FLOW
KICK-OFF



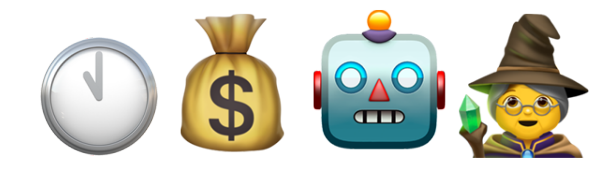
openForBiz()



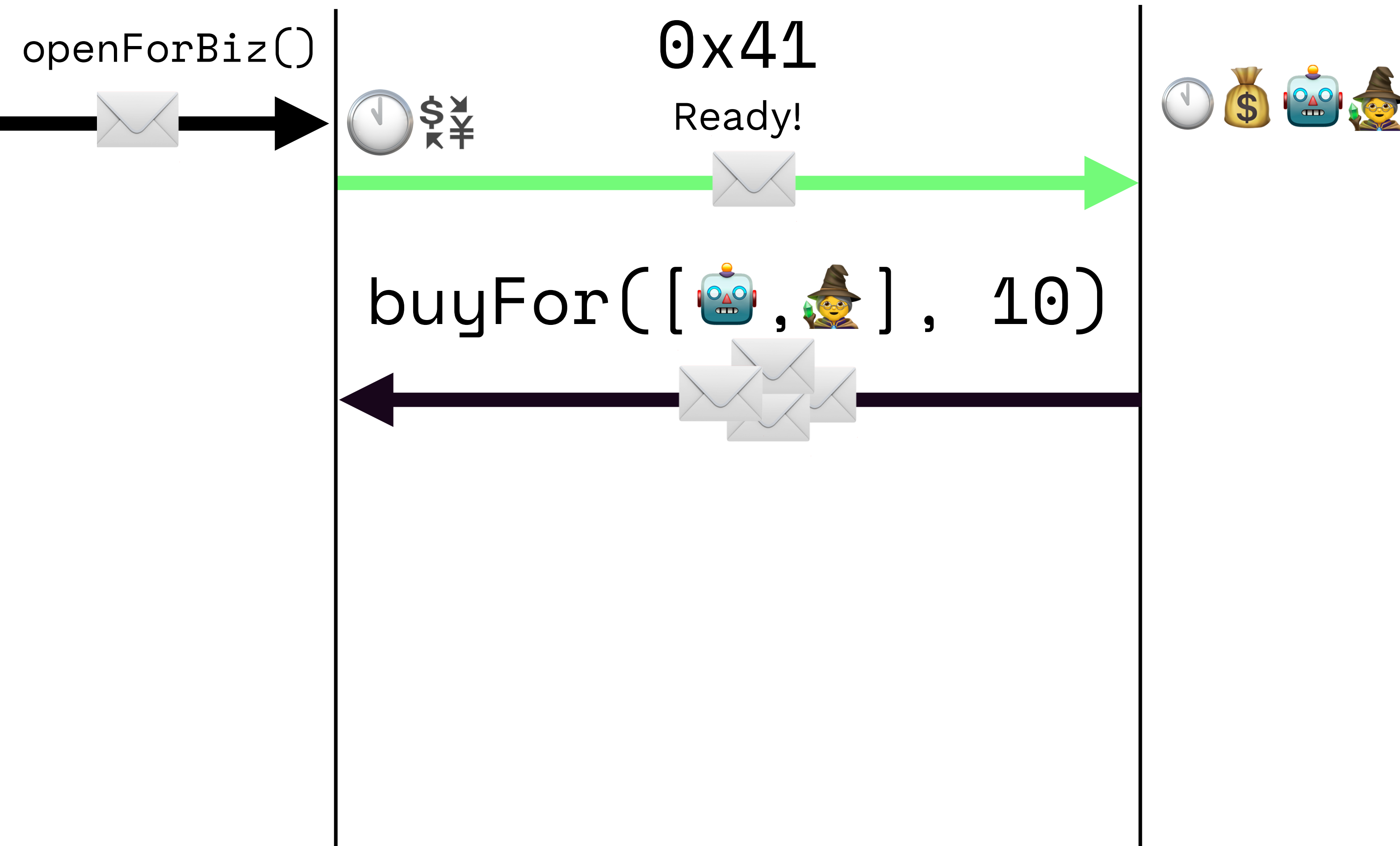
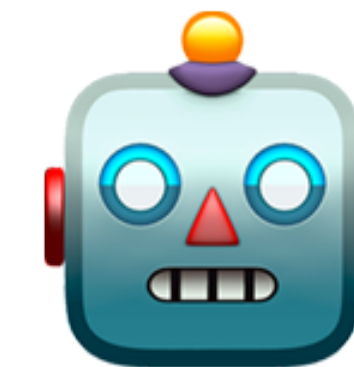
TOY FLOW
KICK-OFF



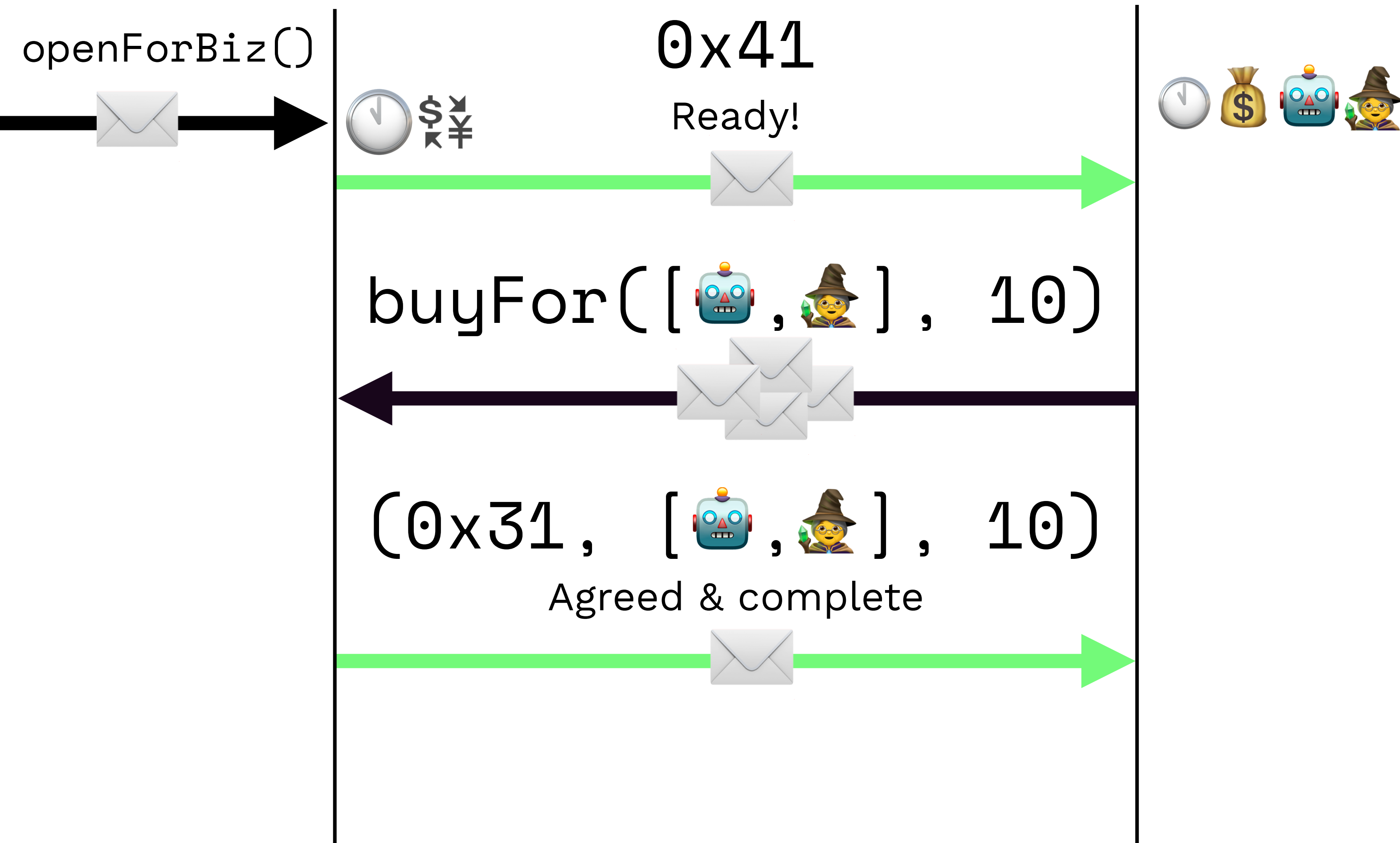
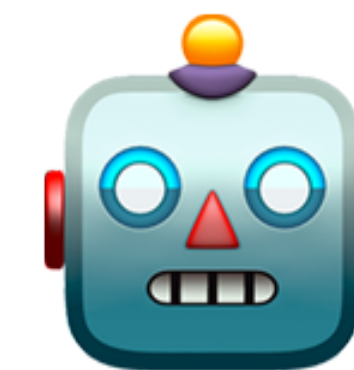
0x41
Ready!



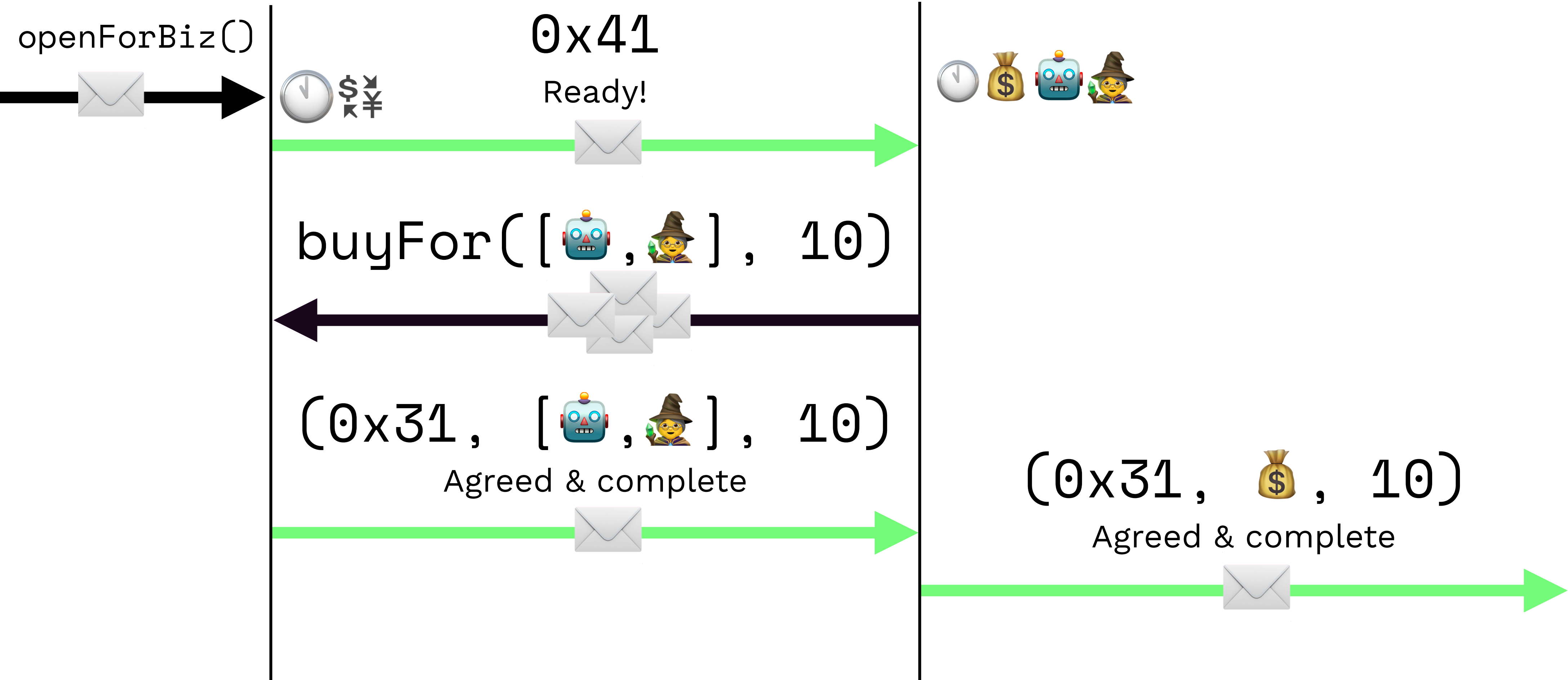
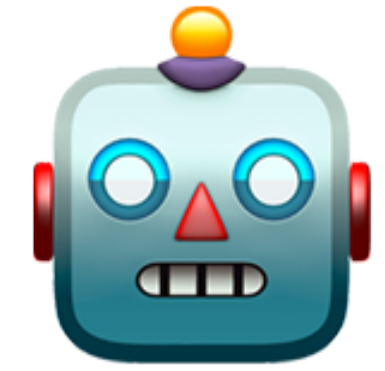
TOY FLOW
KICK-OFF



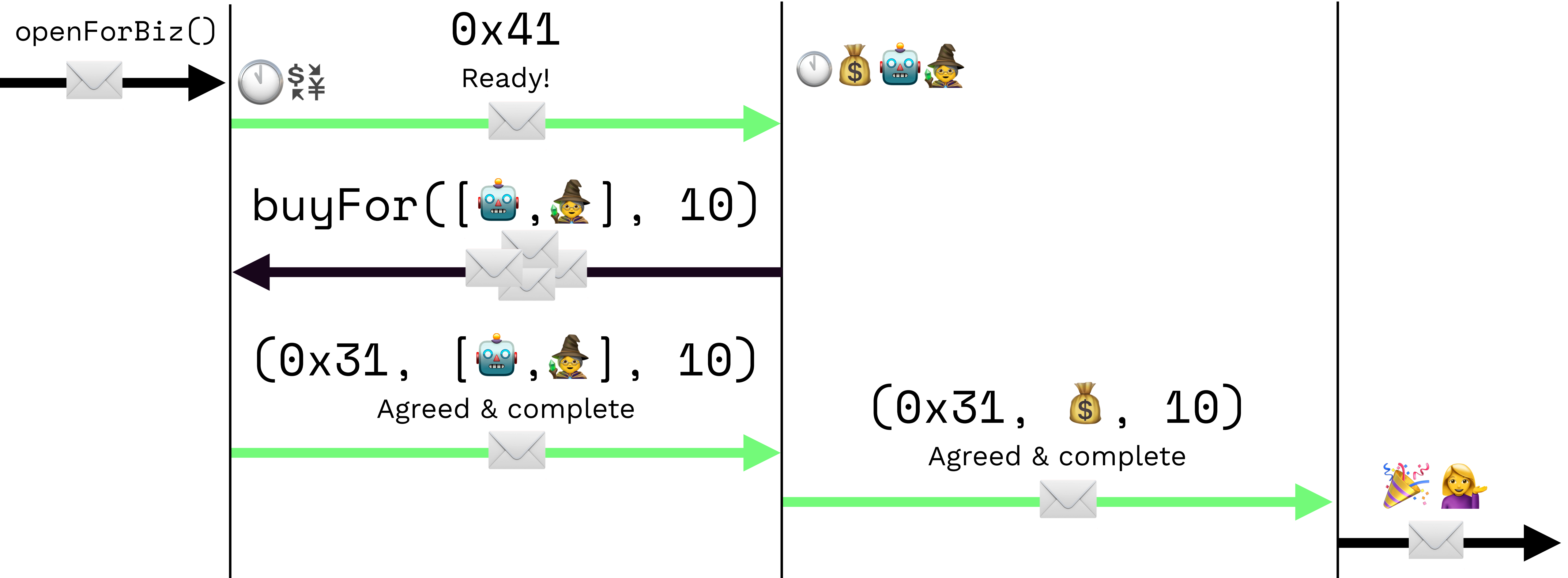
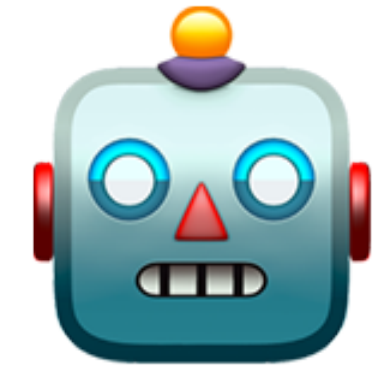
TOY FLOW
KICK-OFF



TOY FLOW
KICK-OFF



TOY FLOW
KICK-OFF



STATELESS SERVICE CONTRACTS

STATELESS SERVICE CONTRACTS

AUDITED, EFFICIENT LIBRARY CODE ✨

STATELESS SERVICE CONTRACTS
SAFE, AUDITED LIBRARIES

STATELESS SERVICE CONTRACTS SAFE, AUDITED LIBRARIES

- Overriding functions and redeploying can be dangerous
- On-chain libraries can be verified & trusted
- Why redeploy SafeMath?
 - It's impractical when **JUMP** is so cheap
- What about more complex, or security-critical functions?

STATELESS SERVICE CONTRACTS

SEPARATION OF CONCERNS

STATELESS SERVICE CONTRACTS

SEPARATION OF CONCERNS

MyToken

```
name = "MyToken"  
totalSupply = 1000000  
...
```

STATELESS SERVICE CONTRACTS SEPARATION OF CONCERNS

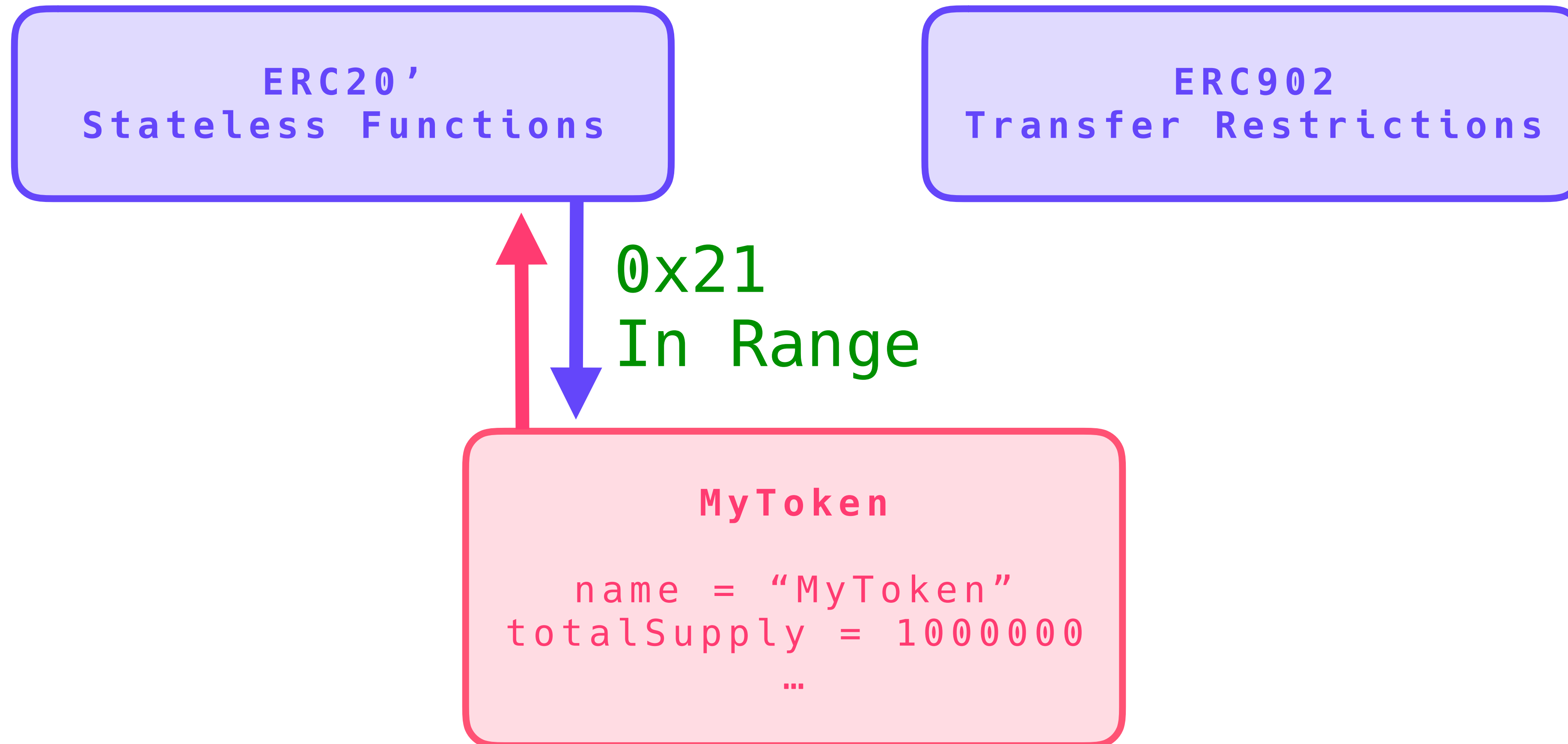
ERC20'
Stateless Functions

ERC902
Transfer Restrictions

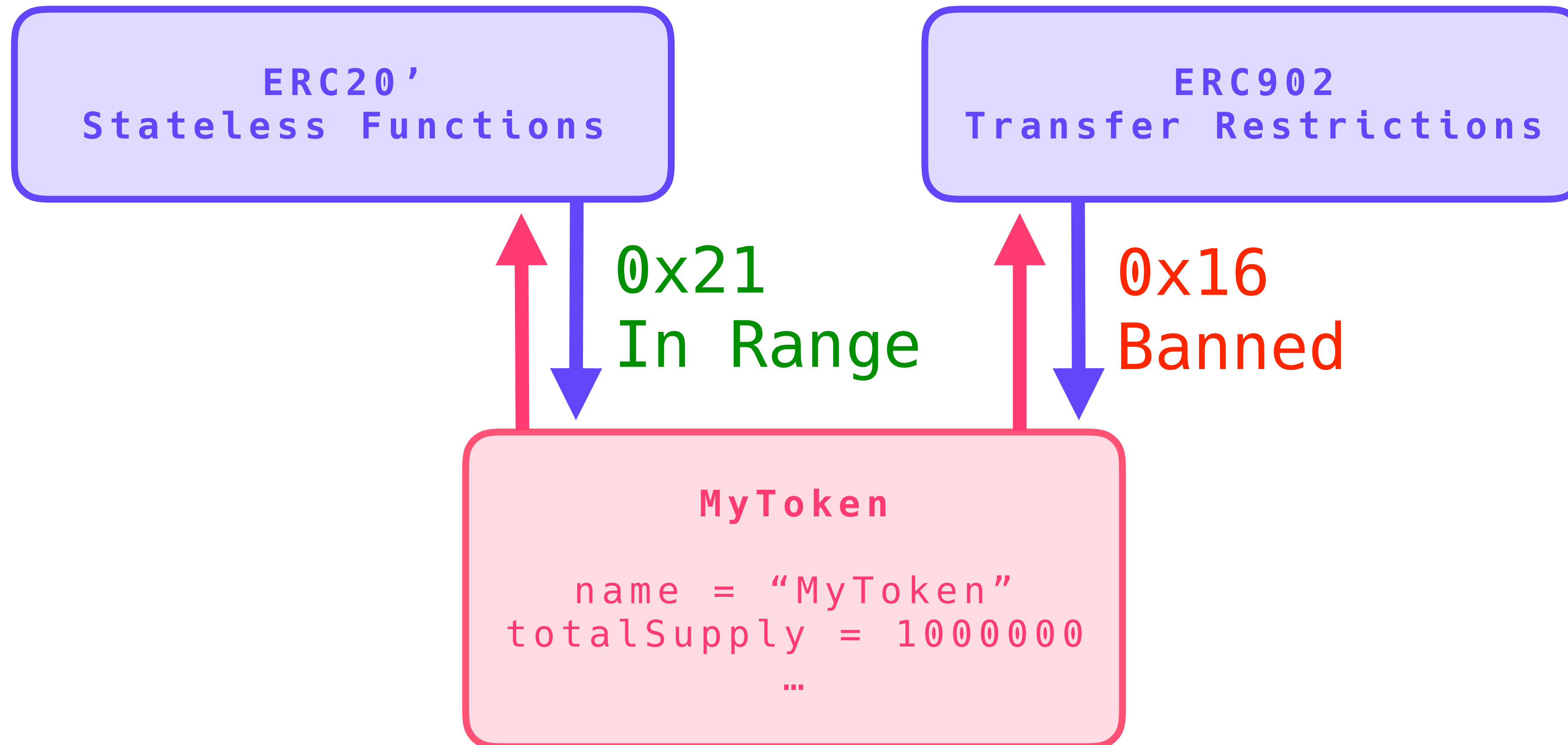
MyToken

```
name = "MyToken"  
totalSupply = 1000000  
...
```


STATELESS SERVICE CONTRACTS SEPARATION OF CONCERNS



STATELESS SERVICE CONTRACTS SEPARATION OF CONCERNS



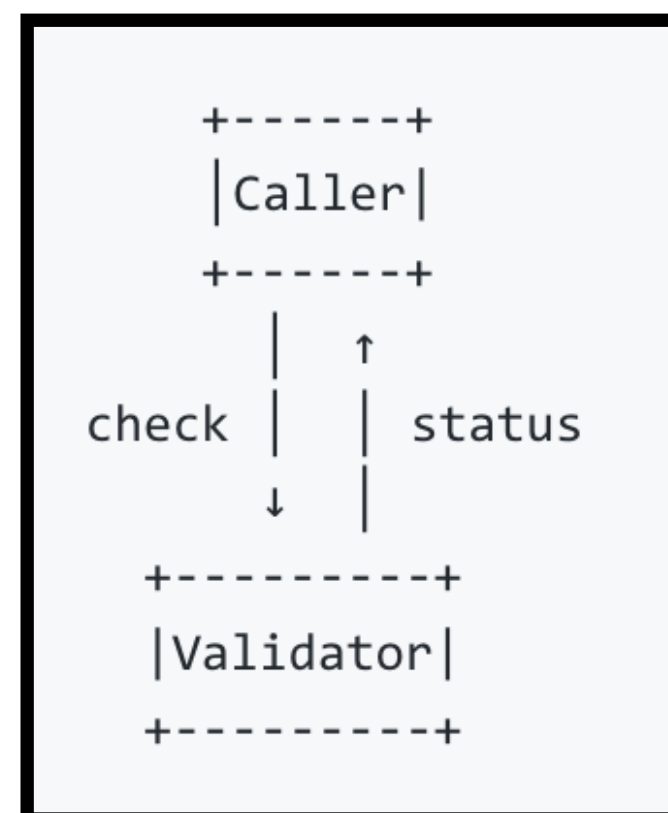
STATEFUL SERVICE CONTRACTS

STATEFUL SERVICE CONTRACTS

DATA CAN BE AN END IN ITSELF 🤖

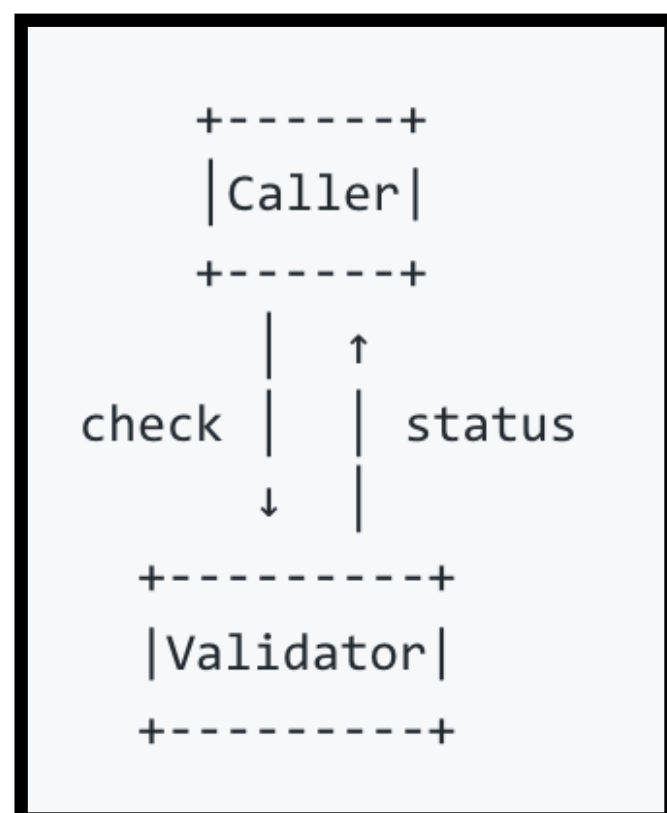
STATEFUL SERVICE CONTRACTS

CROSS-BORDER SECURITY TOKEN



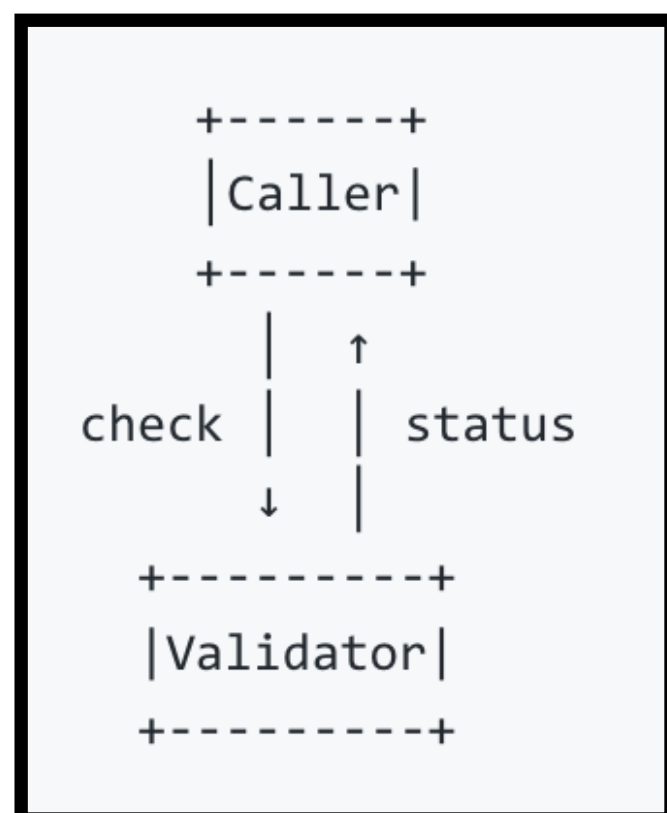
STATEFUL SERVICE CONTRACTS

CROSS-BORDER SECURITY TOKEN



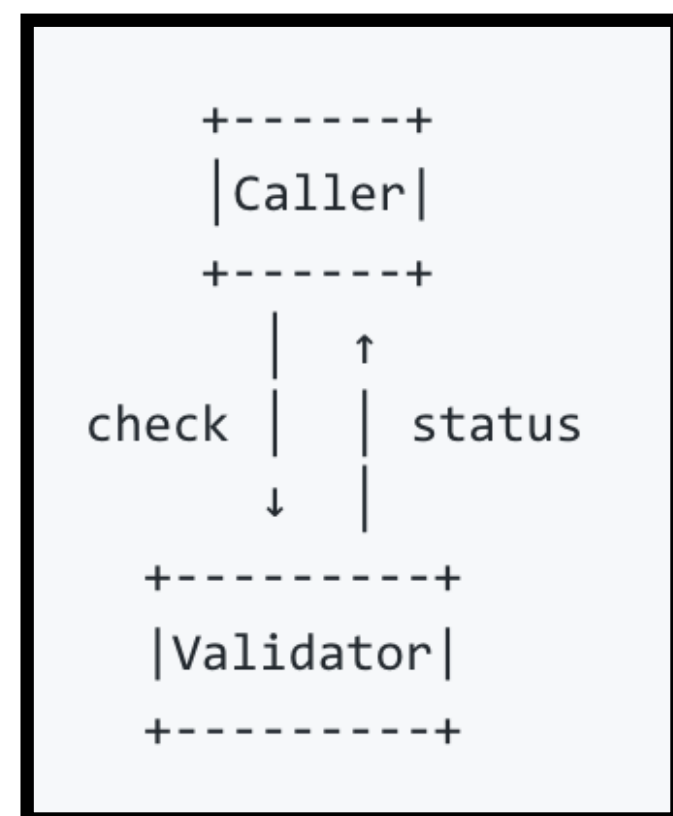
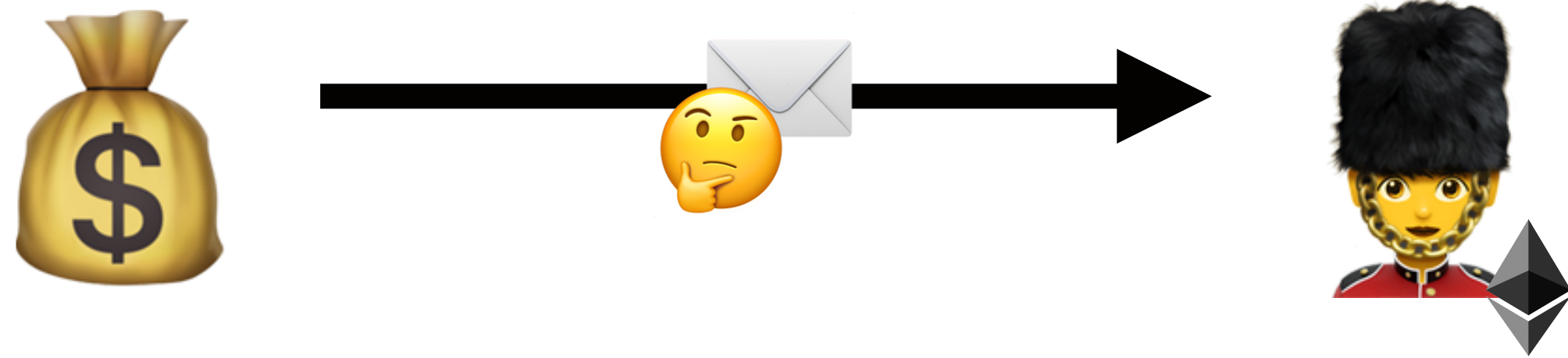
STATEFUL SERVICE CONTRACTS

CROSS-BORDER SECURITY TOKEN

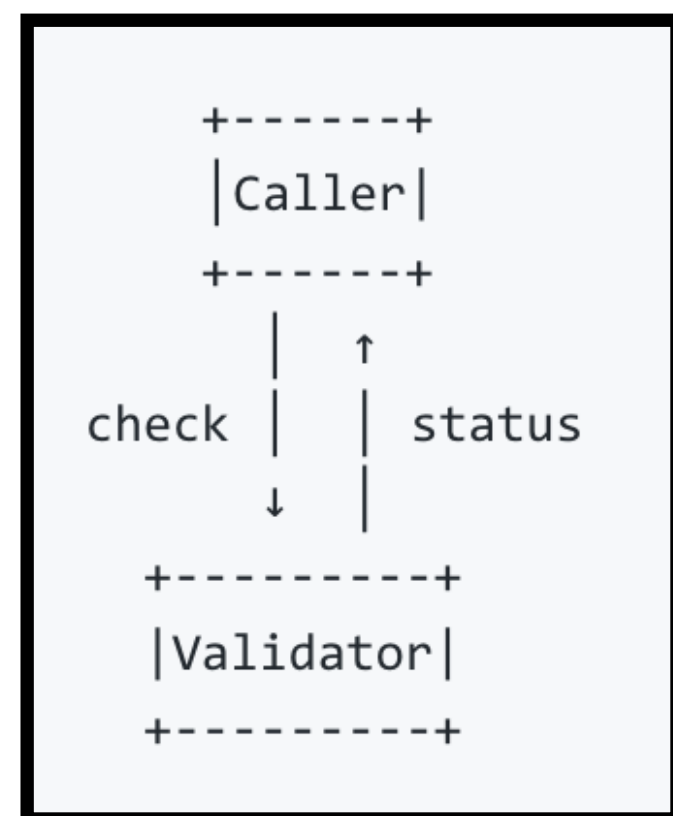
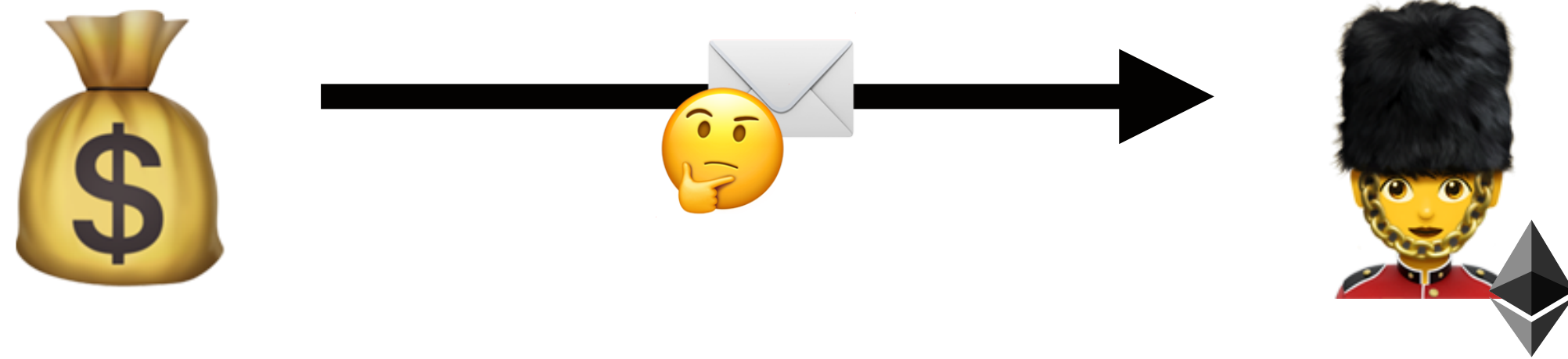


STATEFUL SERVICE CONTRACTS

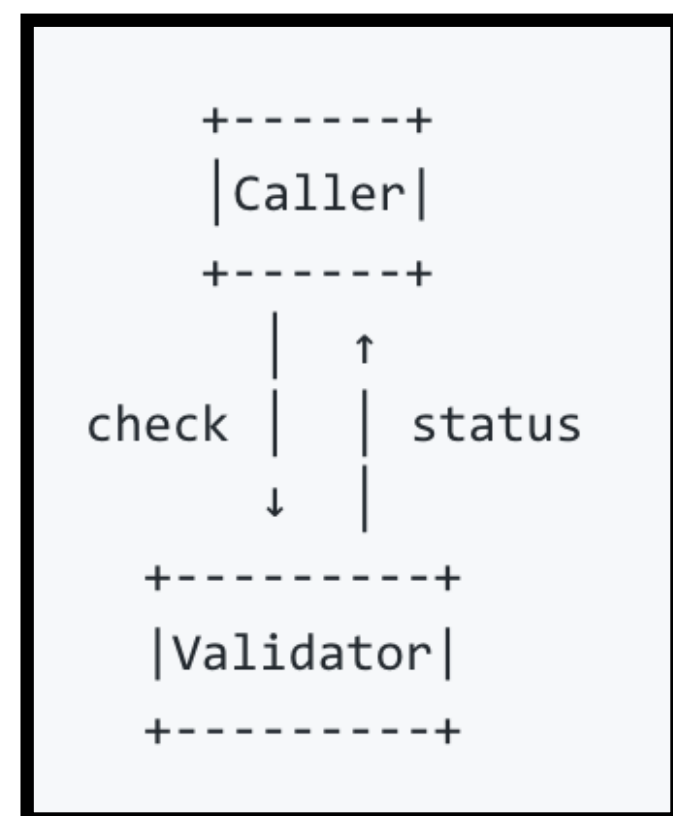
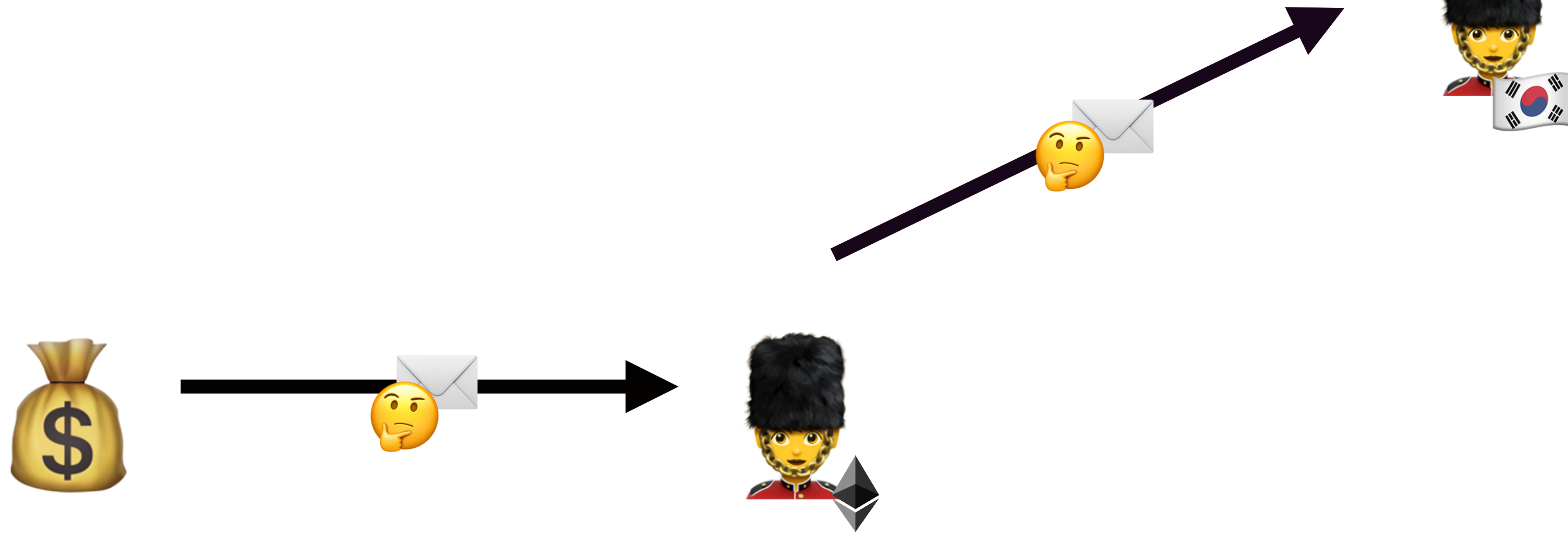
CROSS-BORDER SECURITY TOKEN



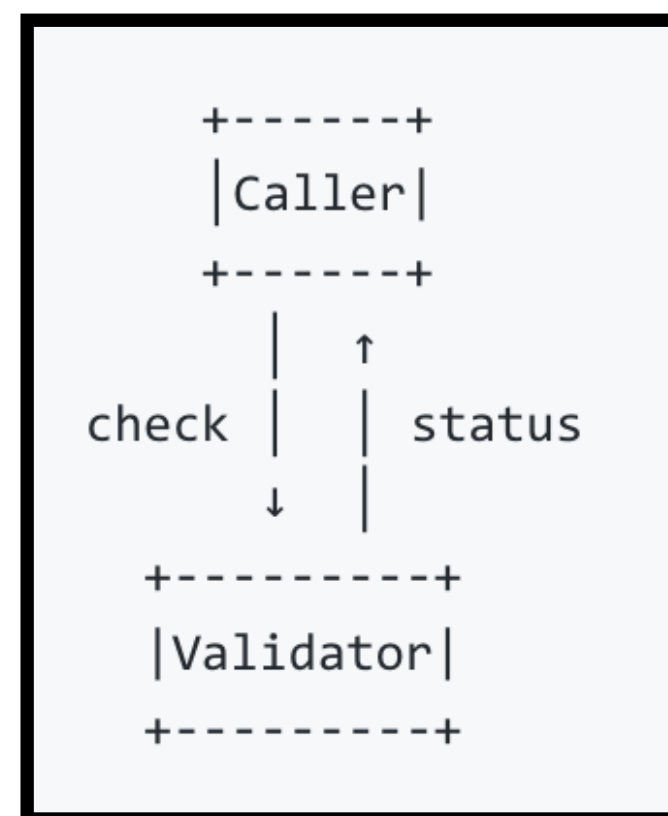
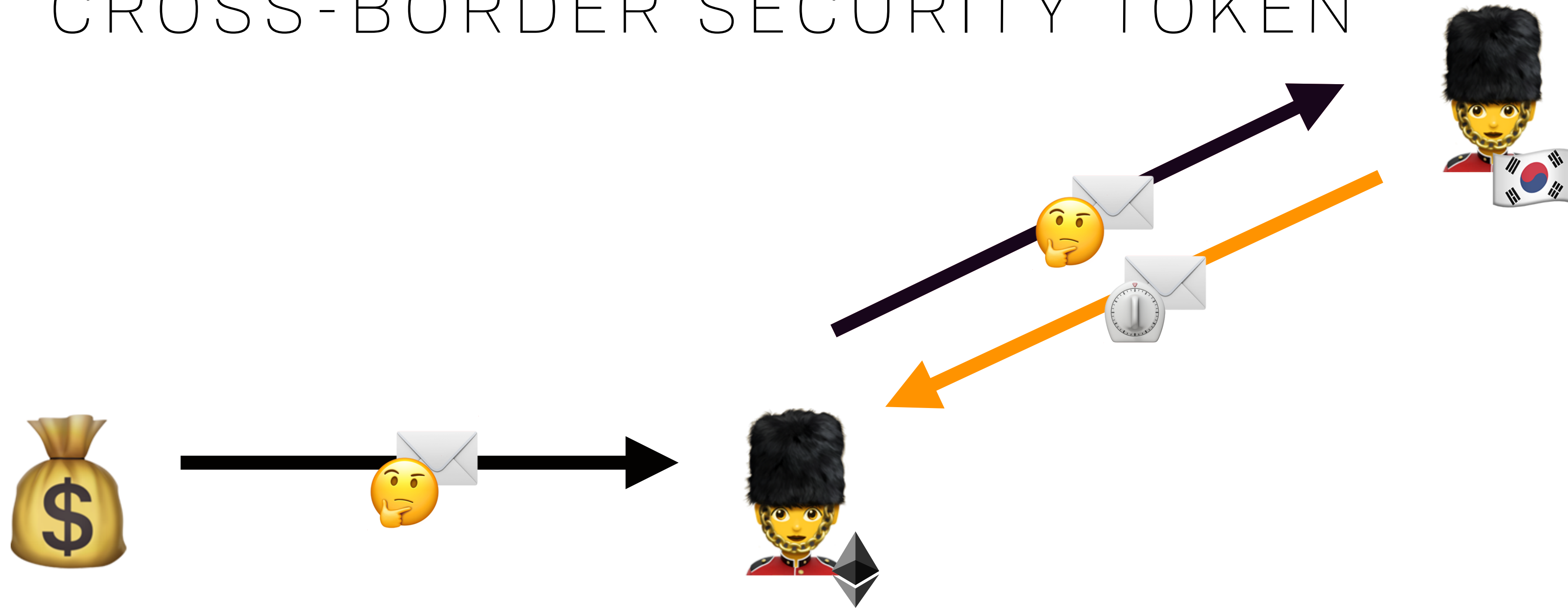
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



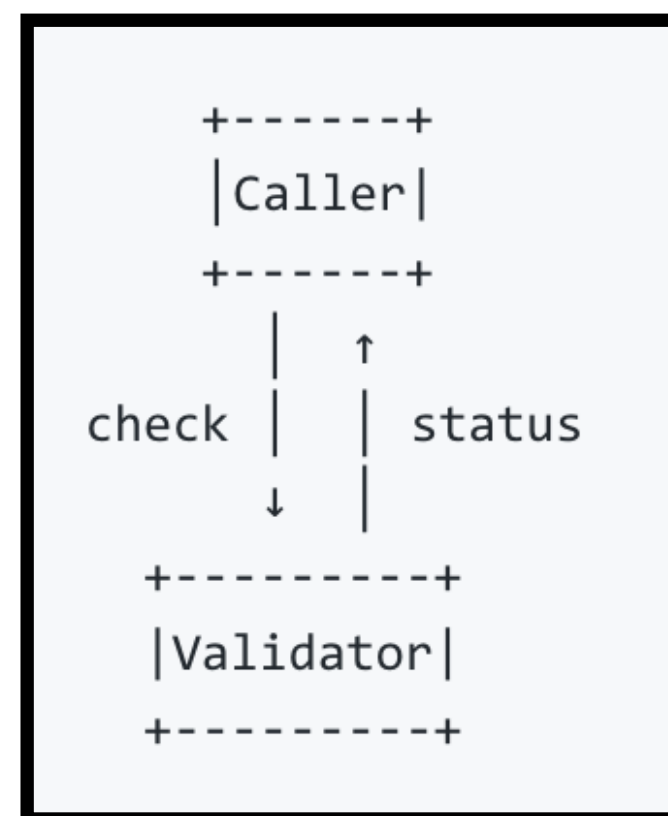
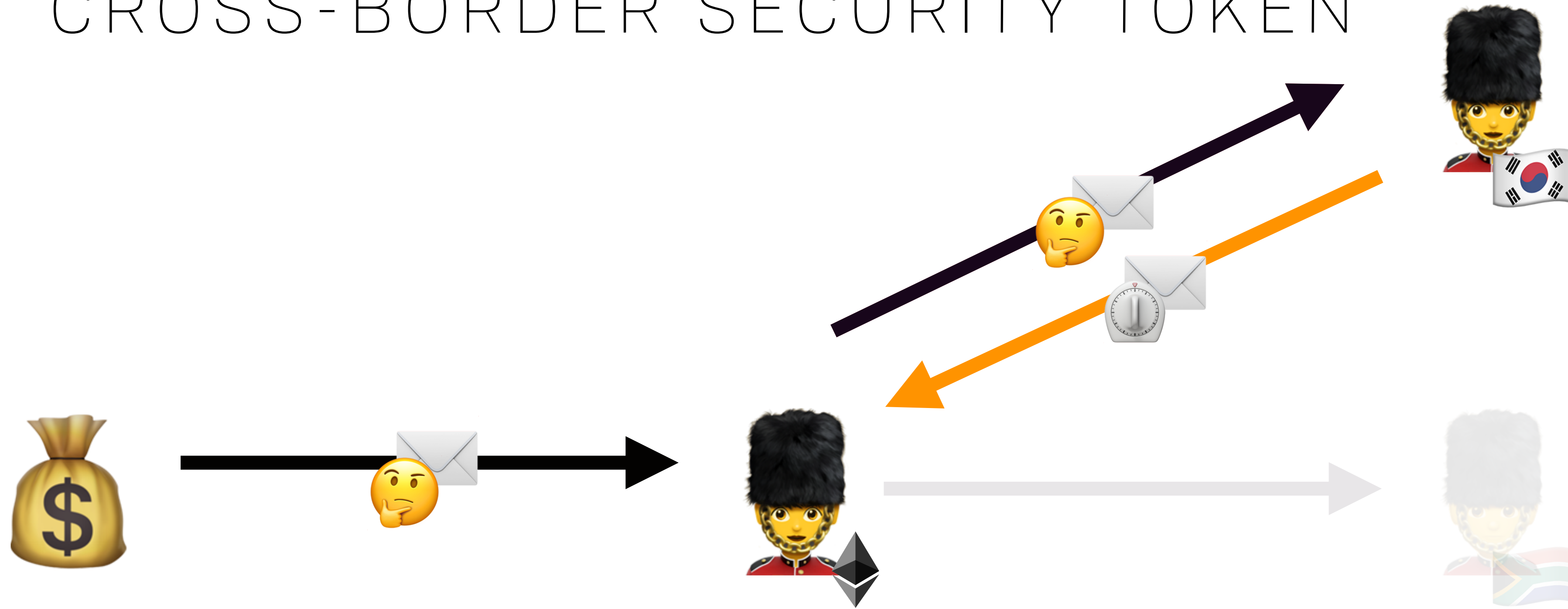
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



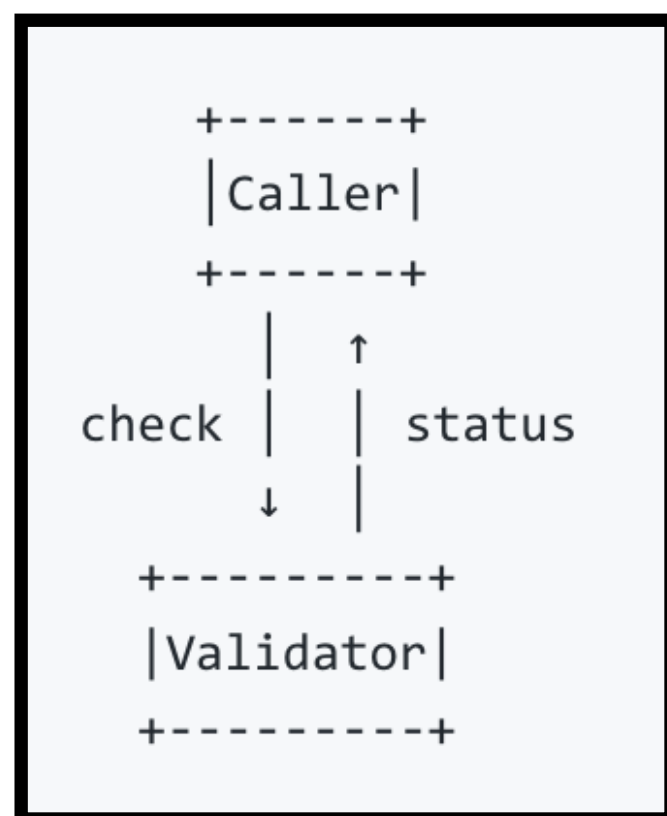
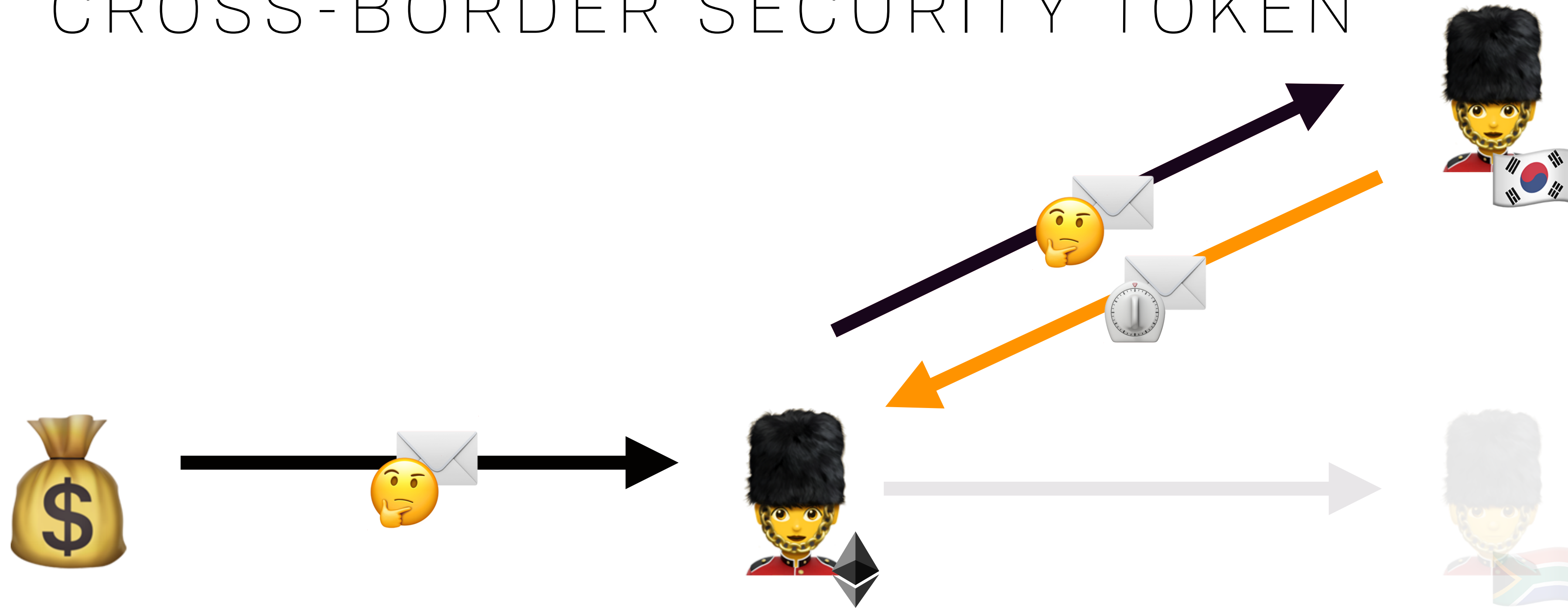
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



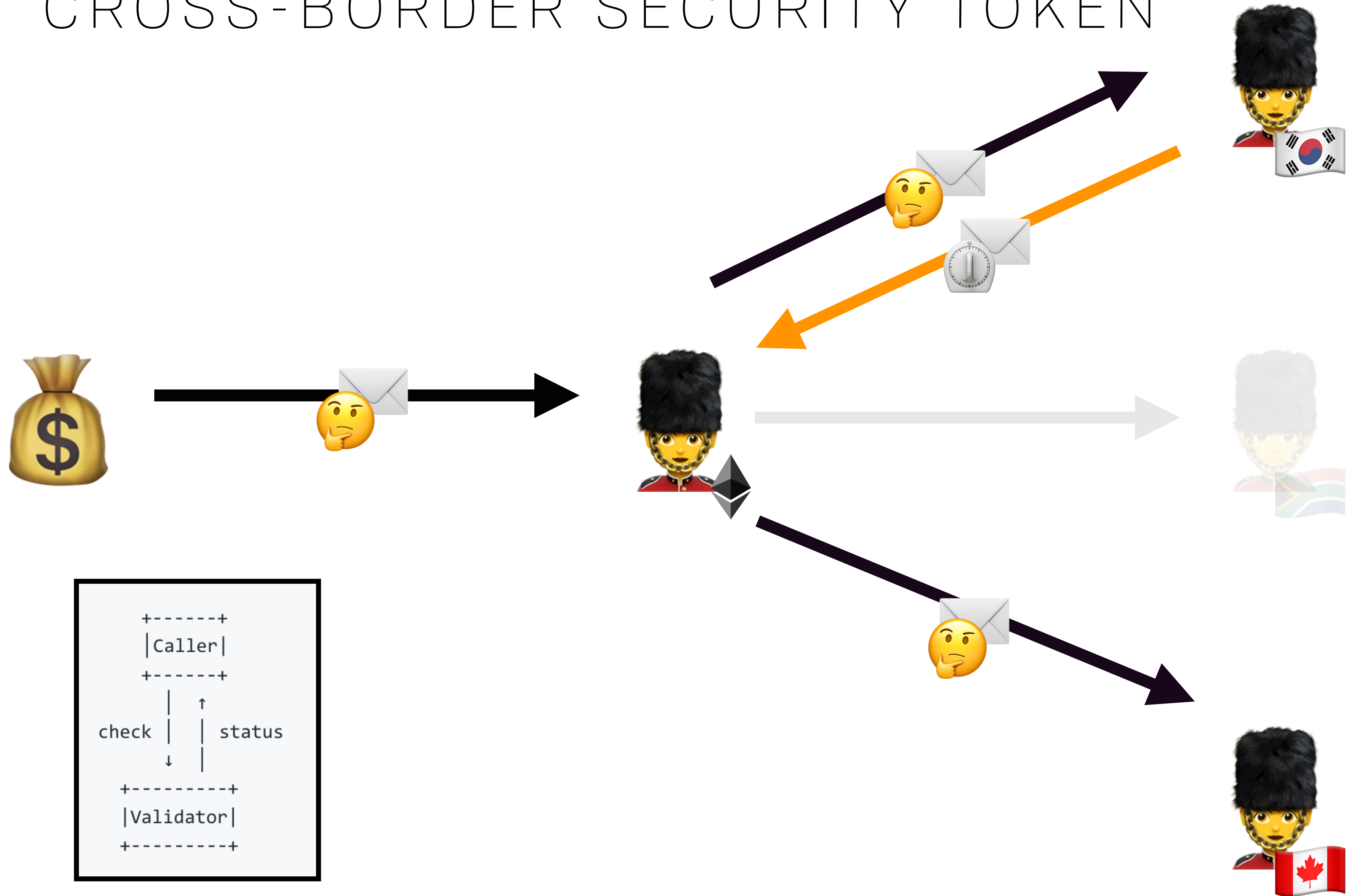
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



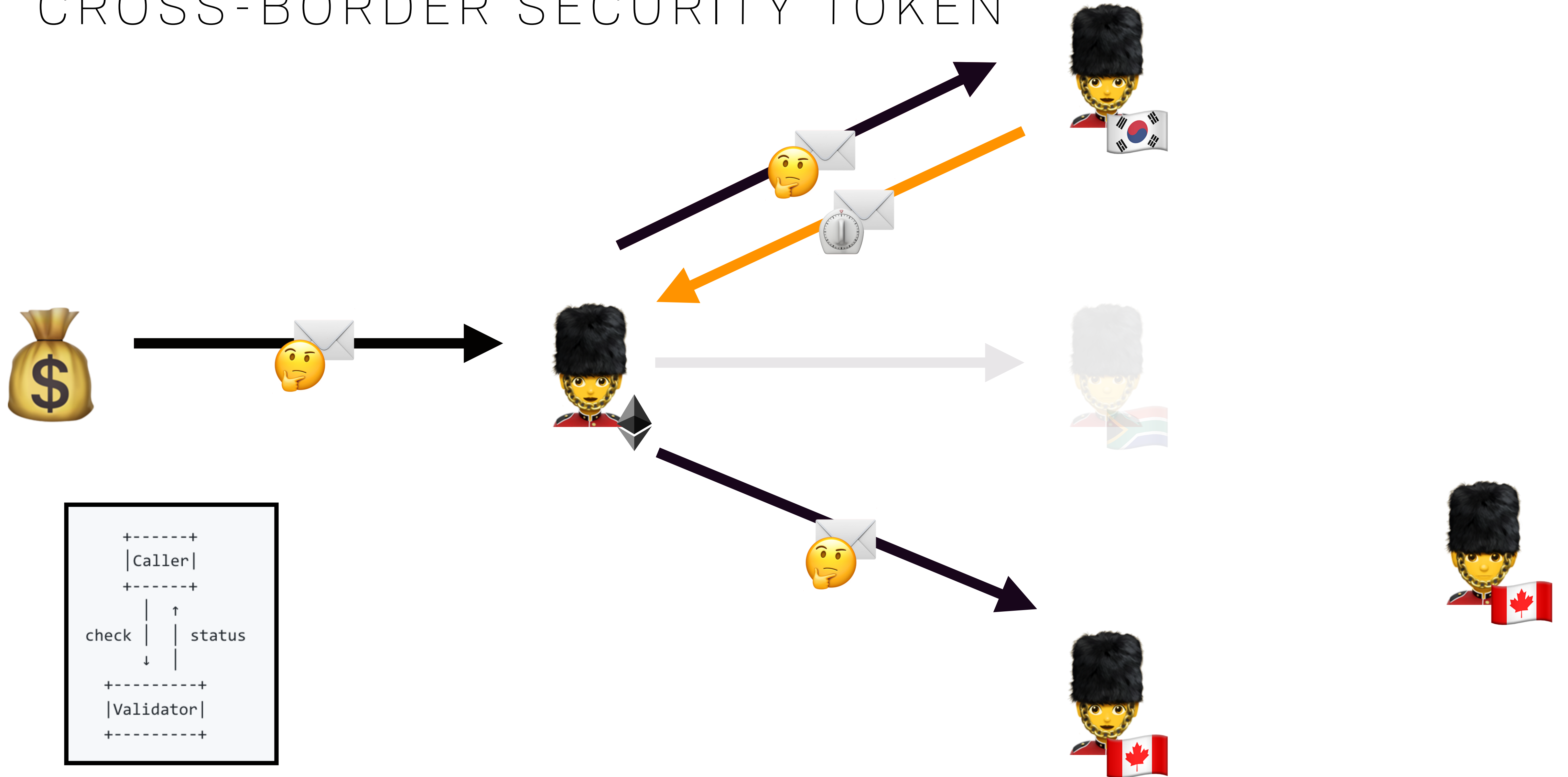
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



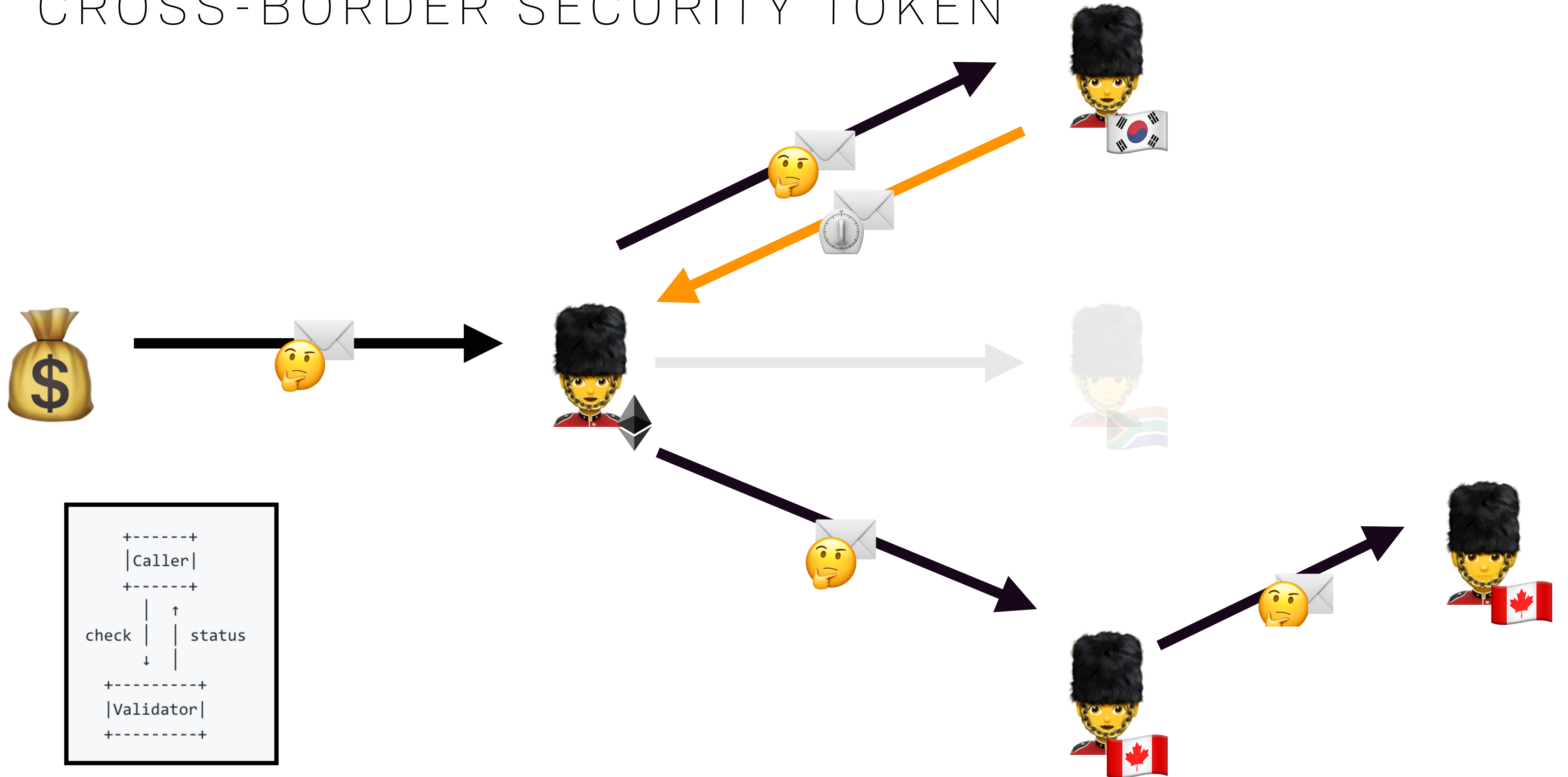
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



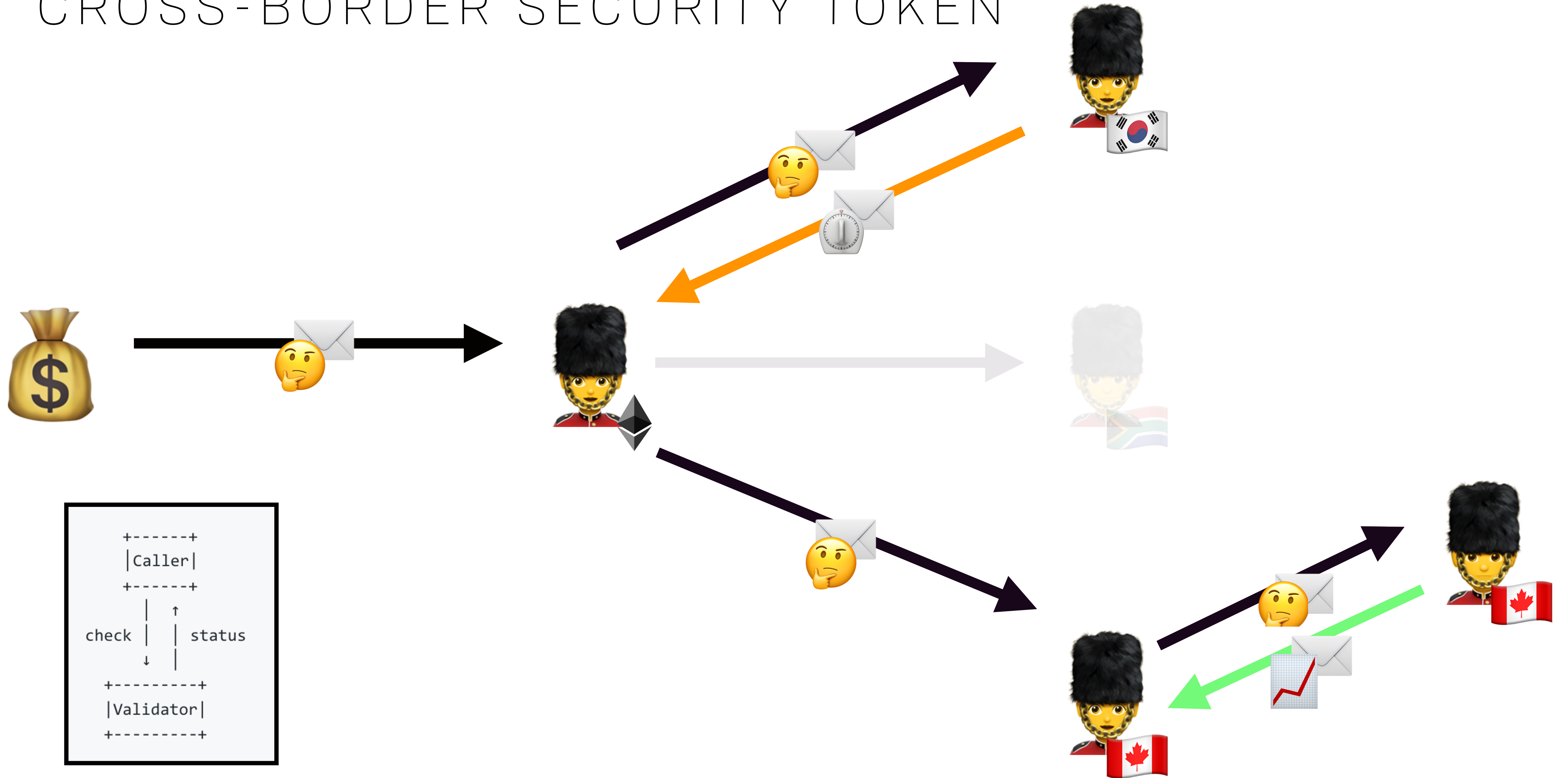
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



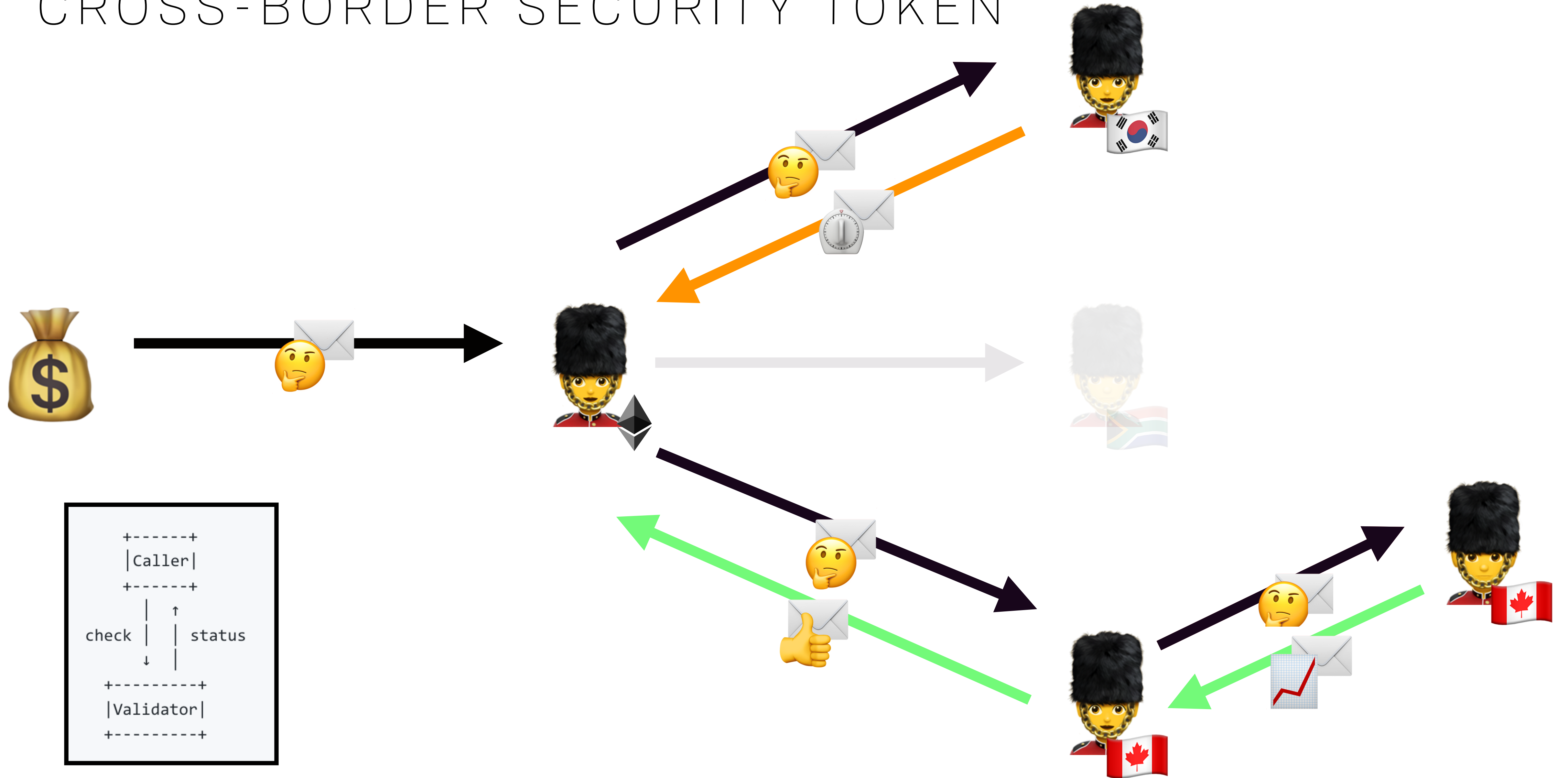
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



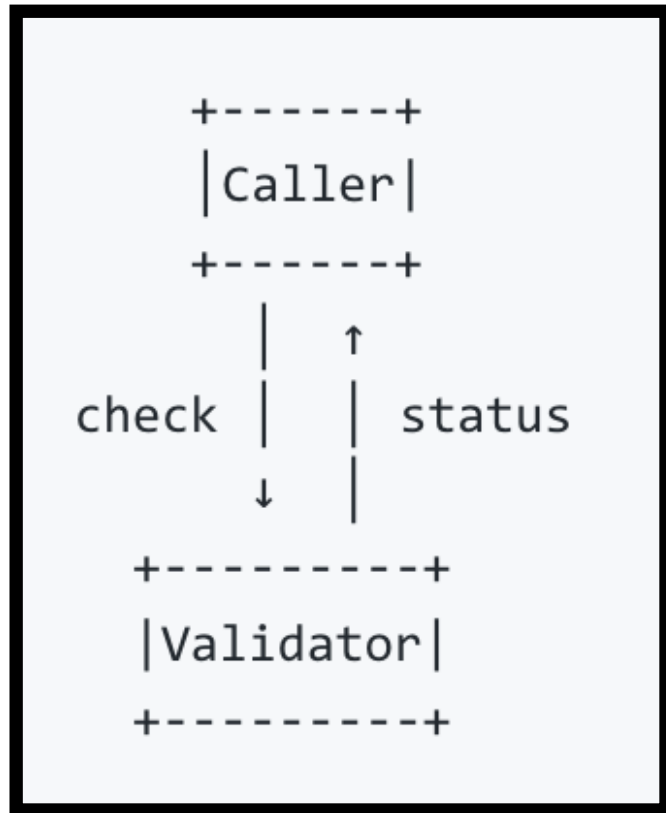
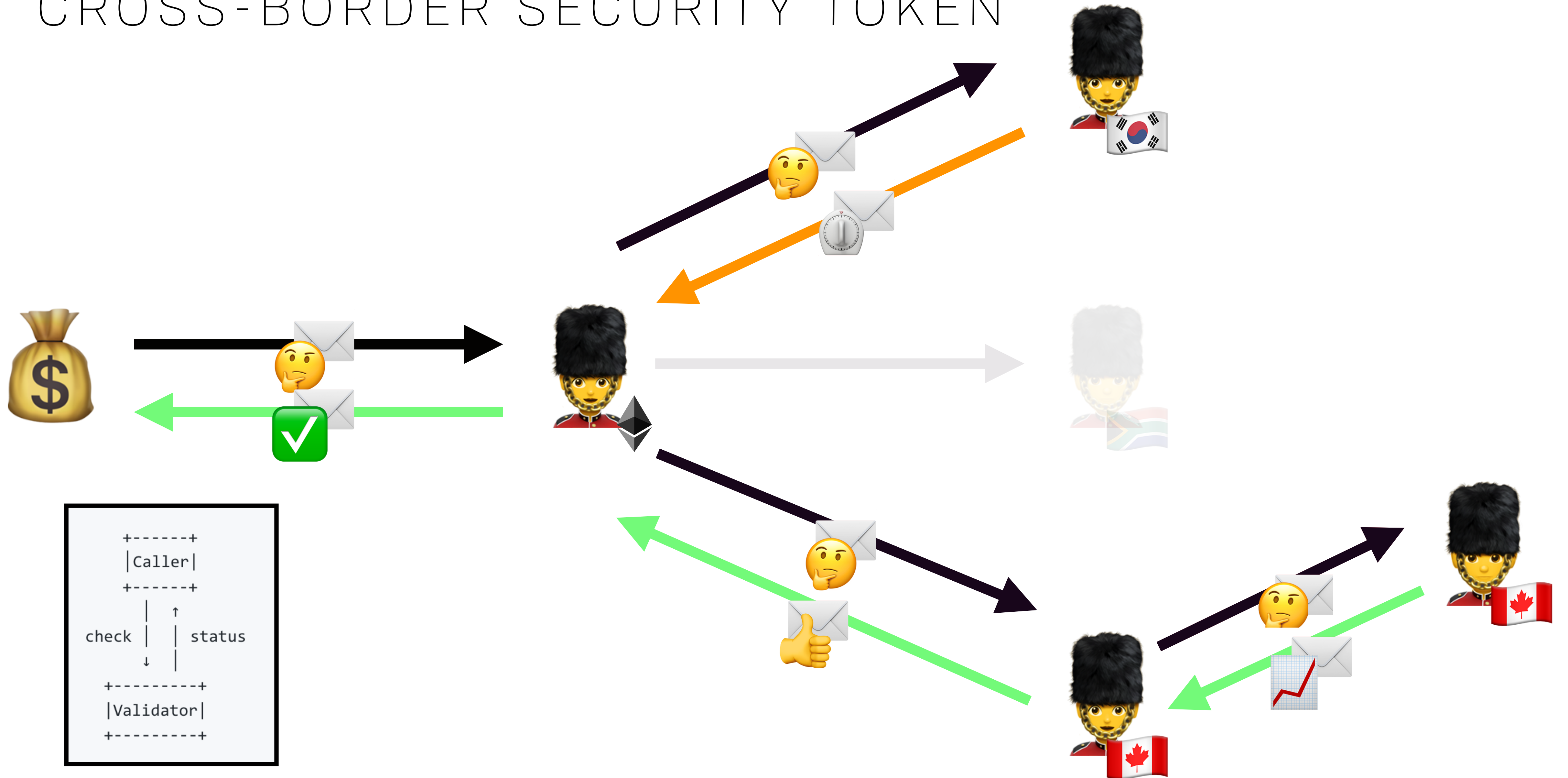
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



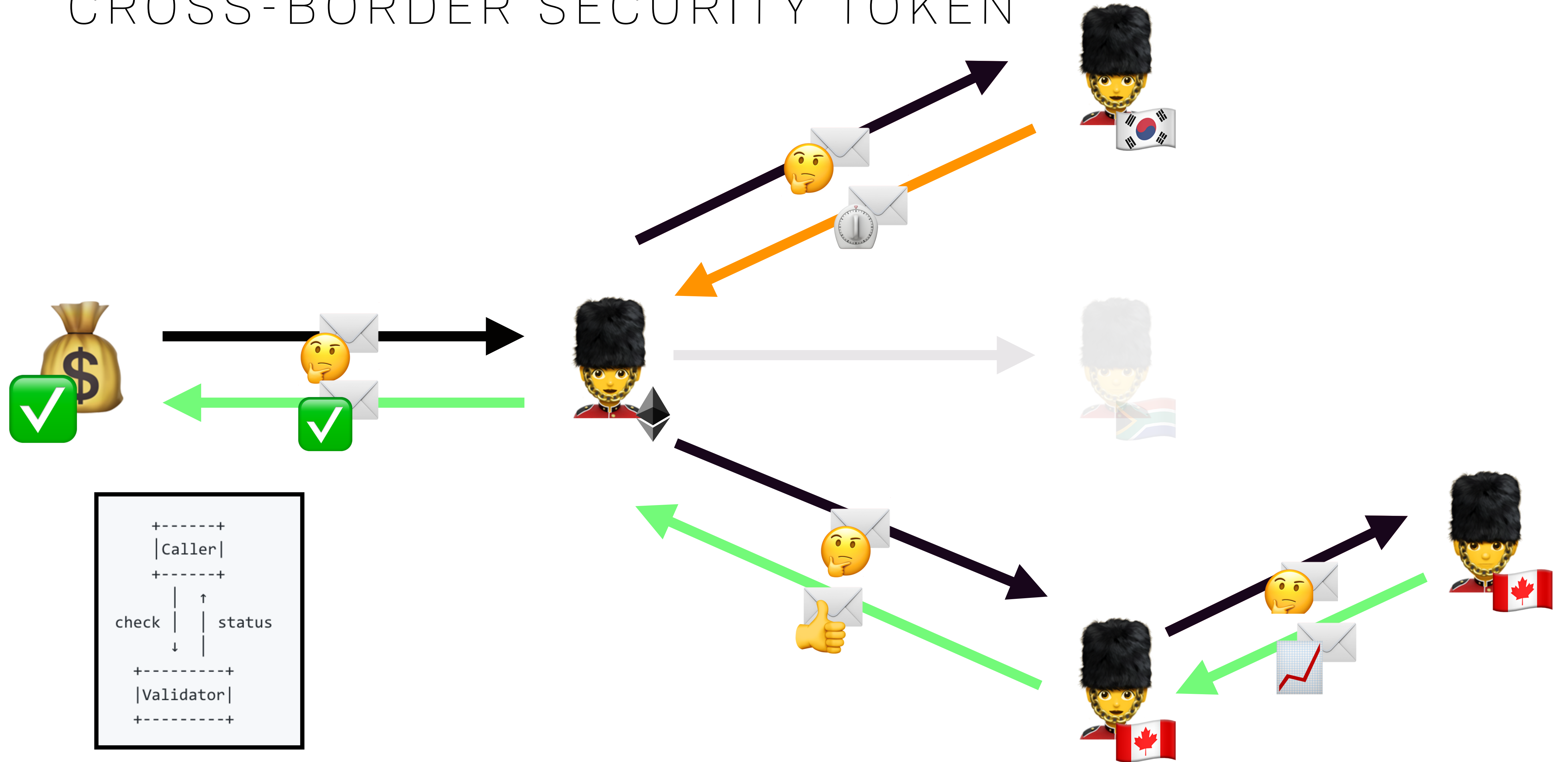
STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN





STATEFUL SERVICE CONTRACTS CROSS-BORDER SECURITY TOKEN



MAKING MONEY

MAKING MONEY

 *BE CALLED OR BE COLLECTED* 

MAKING MONEY

THOUGHT EXPERIMENT: RENTAL ECONOMY

MAKING MONEY

THOUGHT EXPERIMENT: RENTAL ECONOMY

- Charge for function calls
- Not unheard of today, but less common
- Audits are expensive
- High value data is... high value
- Incentivize high quality, audited code by paying for it's creation

MAKING MONEY

SMART CONTRACT ECONOMY

MAKING MONEY

SMART CONTRACT ECONOMY

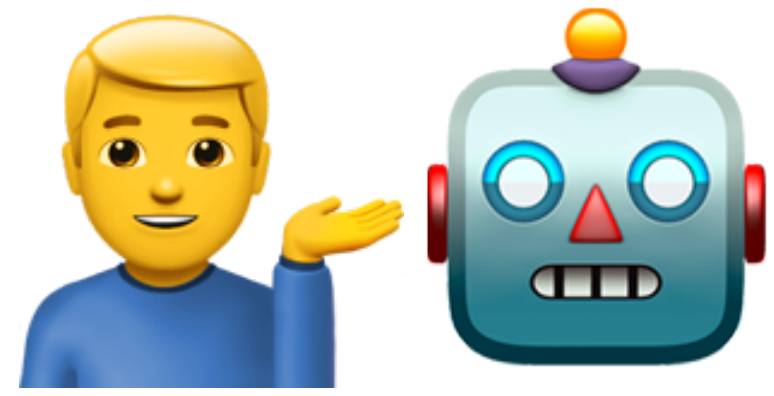
- In a hypothetical state rent future this behaves like an economy
 - Incentivize high-quality contracts that can do more than just survive
 - The ones that don't provide value "go under"
- Per-invocation "lambda-style" charges
- Buy a license on chain
 - Per-contract or per-vendor
 - Time
 - Permanent
 - Number of calls
 - Number of blocks

USE CASE: LOCALIZED FEEDBACK

USE CASE: LOCALIZED FEEDBACK



USER FEEDBACK ON-CHAIN LOCALIZATION



Requestor



LocalizationPreference



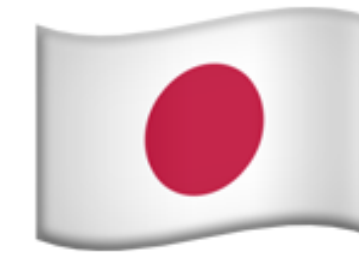
Localization



Localization



Localization



Localization



Localization



Localization

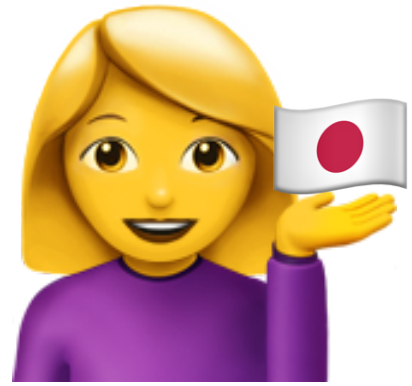


Localization

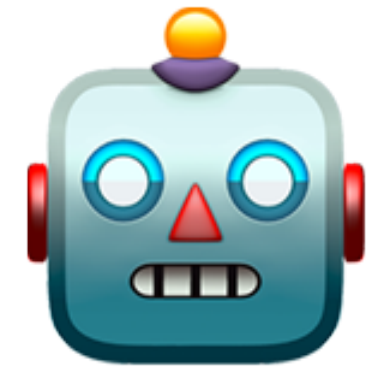
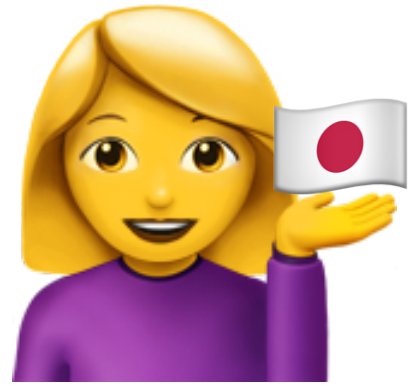
USER FEEDBACK

SUCCESS FLOW

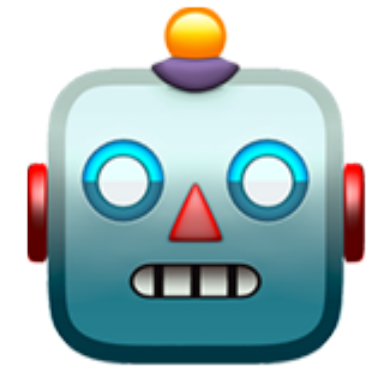
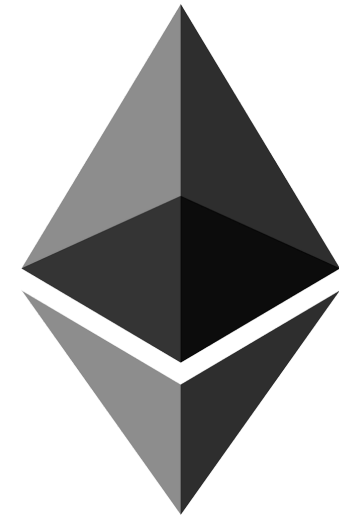
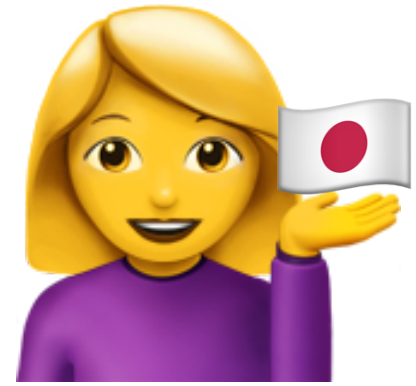
USER FEEDBACK SUCCESS FLOW



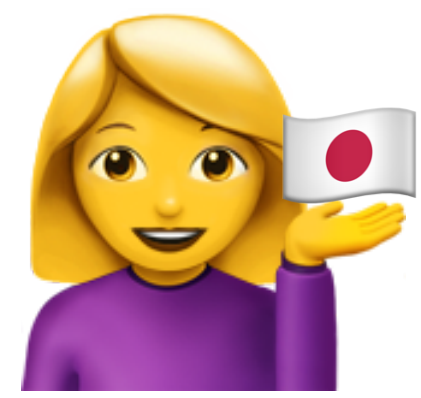
USER FEEDBACK SUCCESS FLOW



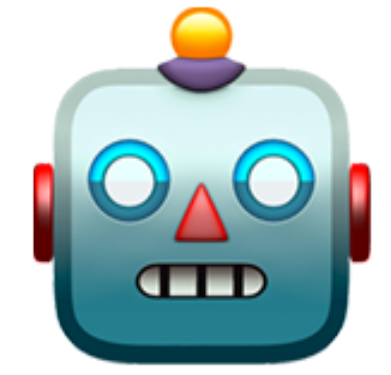
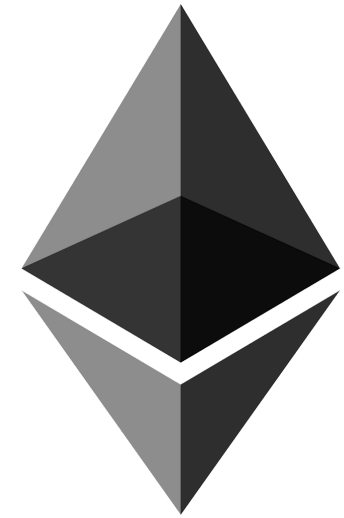
USER FEEDBACK SUCCESS FLOW



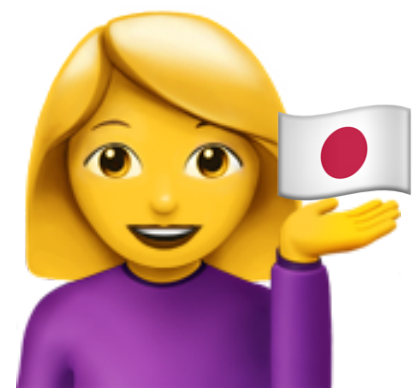
USER FEEDBACK SUCCESS FLOW



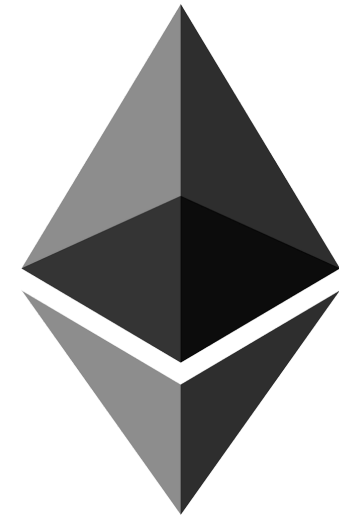
tx



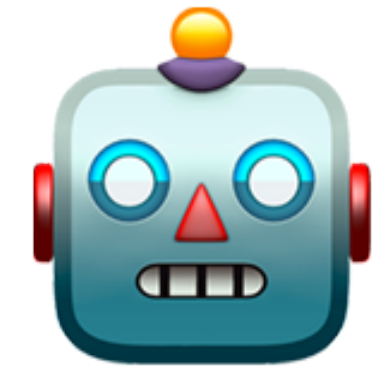
USER FEEDBACK
SUCCESS FLOW



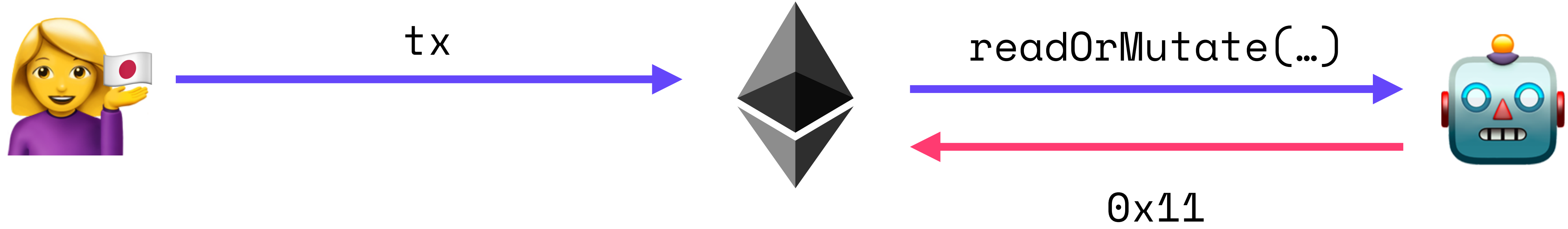
tx



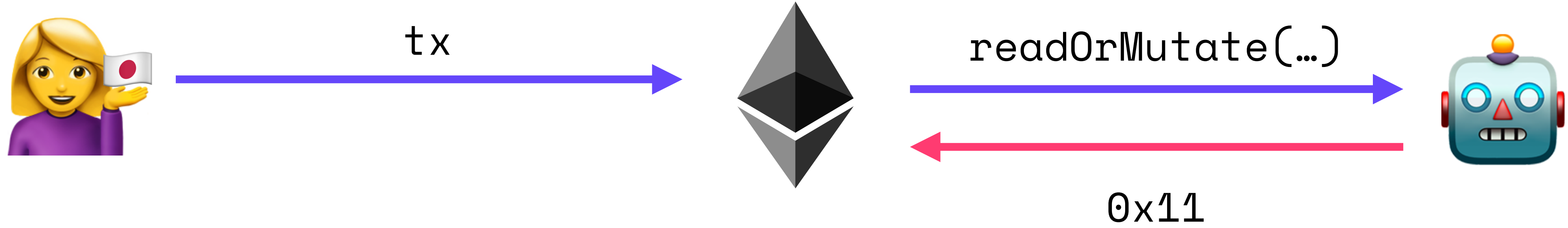
readOrMutate(...)



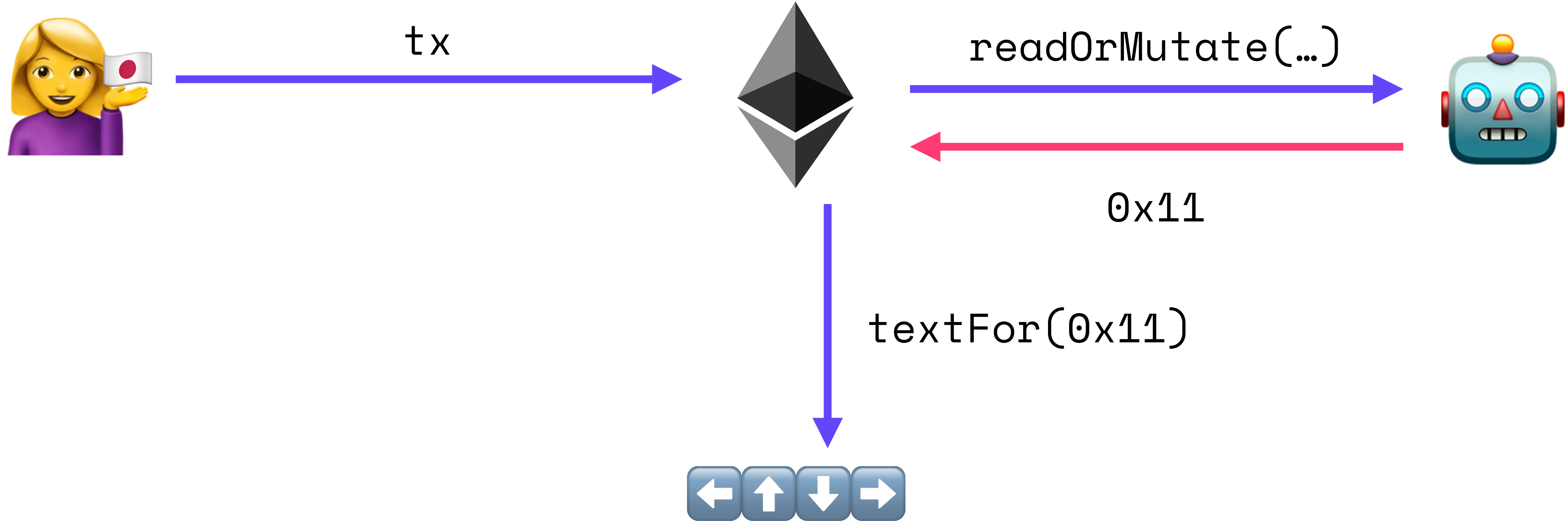
USER FEEDBACK
SUCCESS FLOW



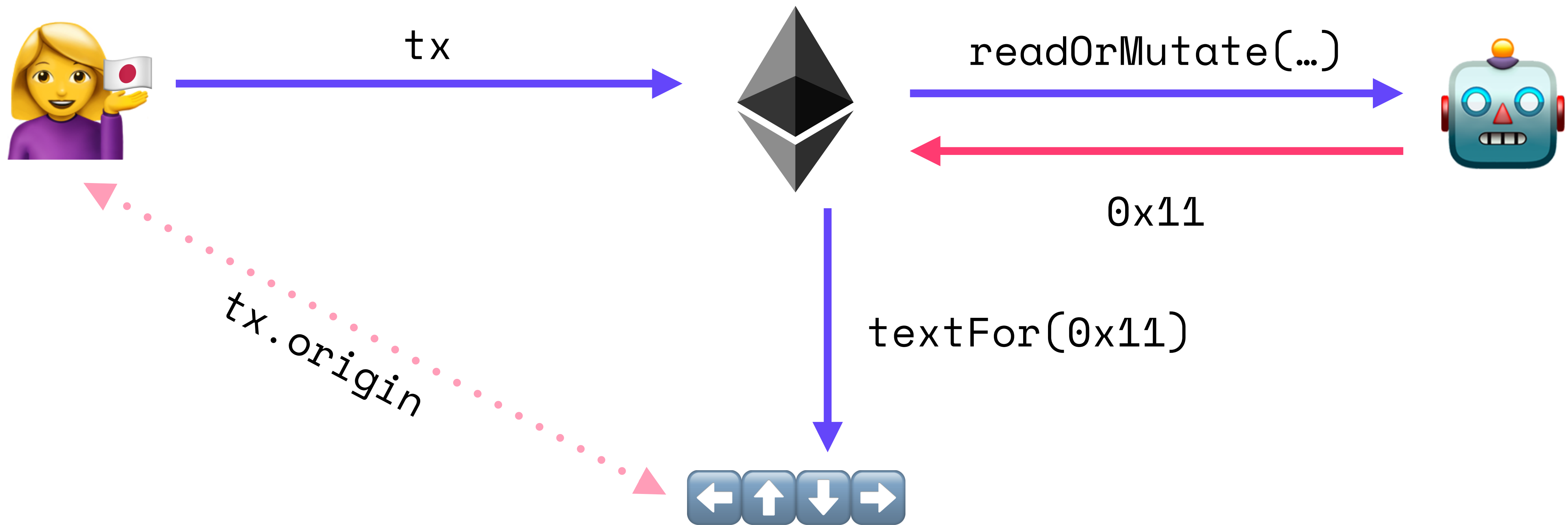
USER FEEDBACK SUCCESS FLOW



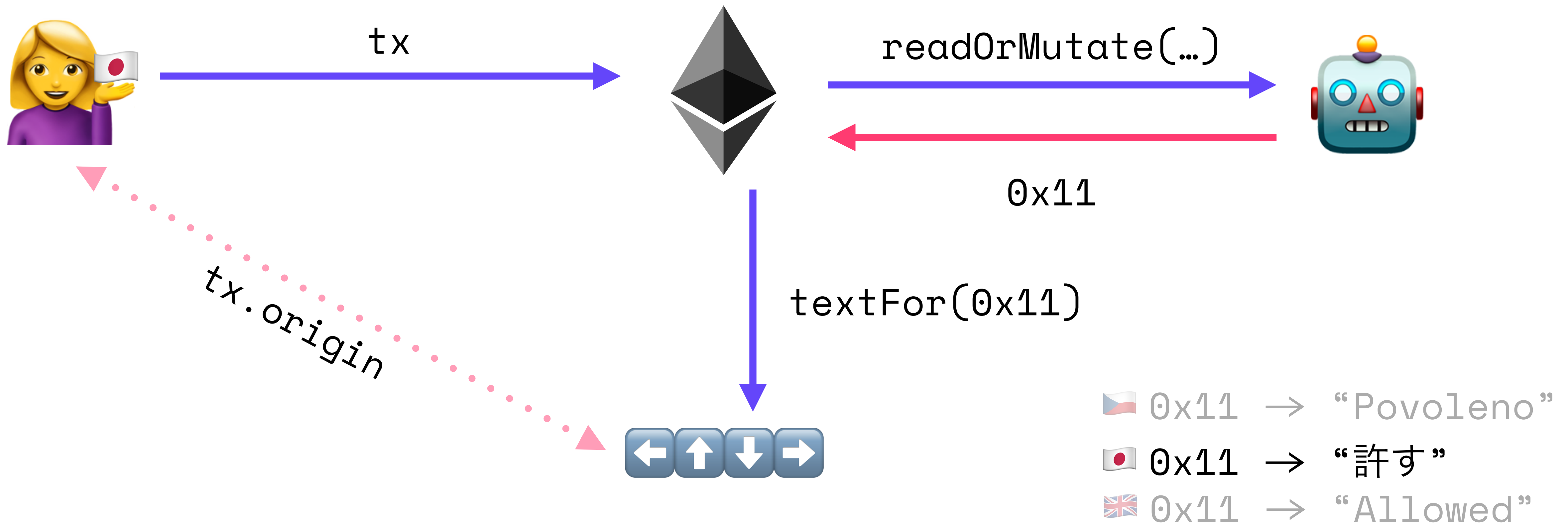
USER FEEDBACK
SUCCESS FLOW



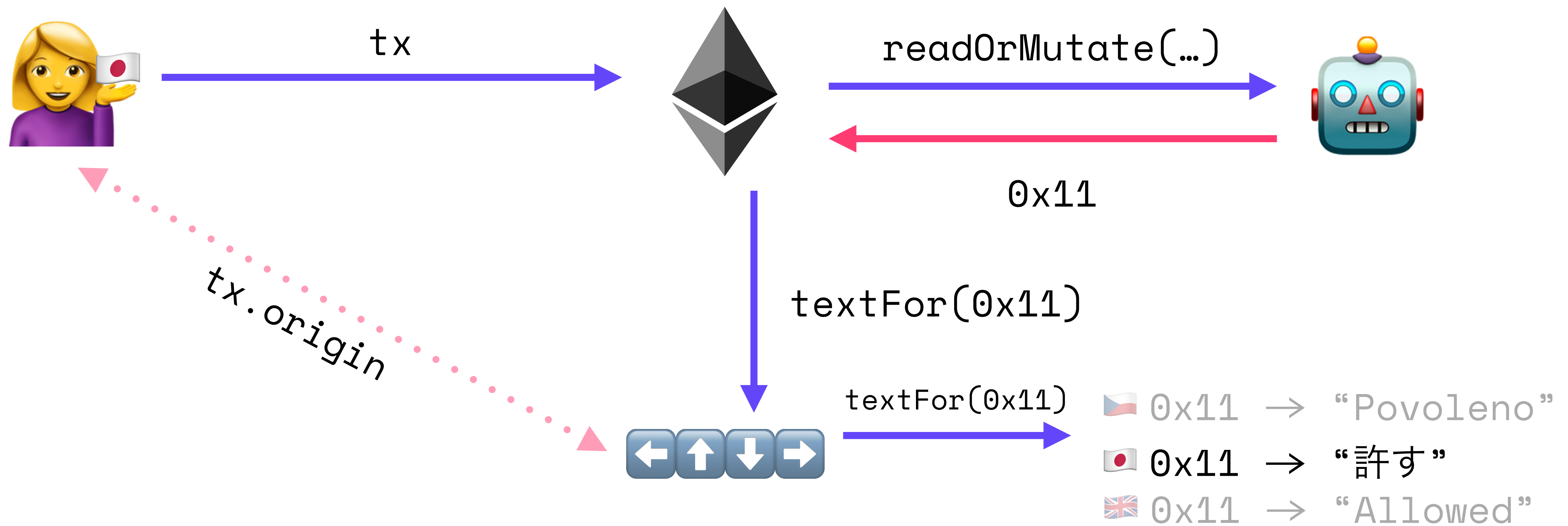
USER FEEDBACK SUCCESS FLOW



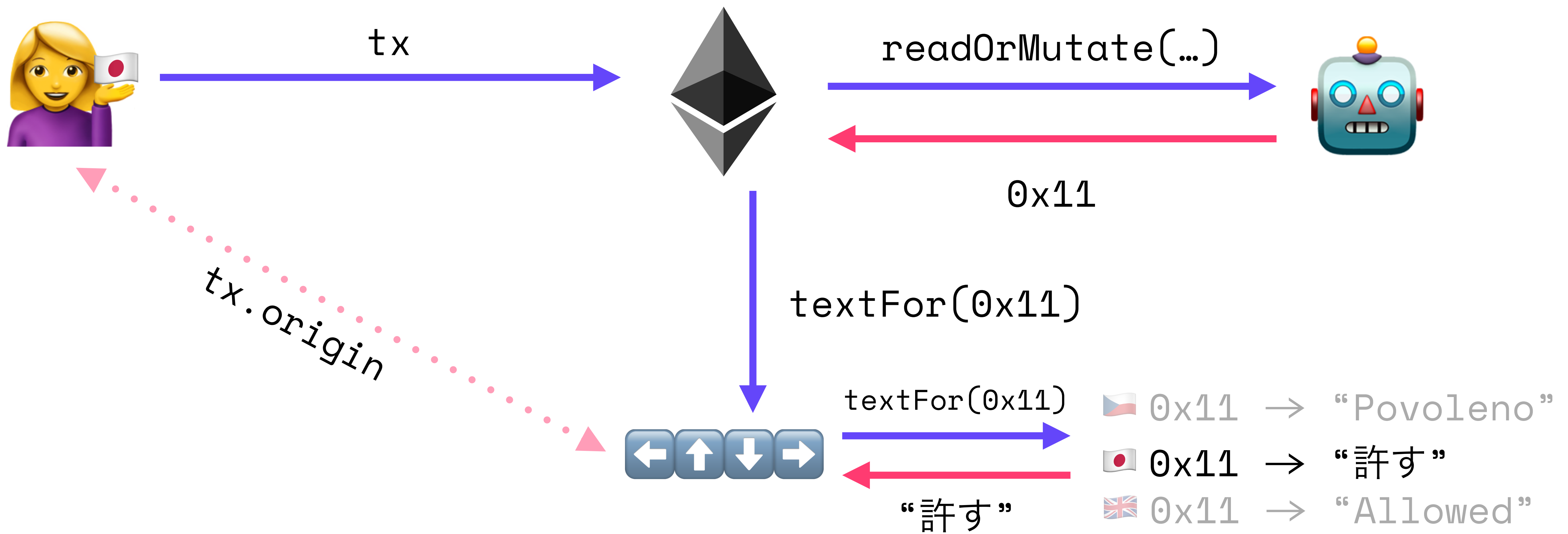
USER FEEDBACK SUCCESS FLOW



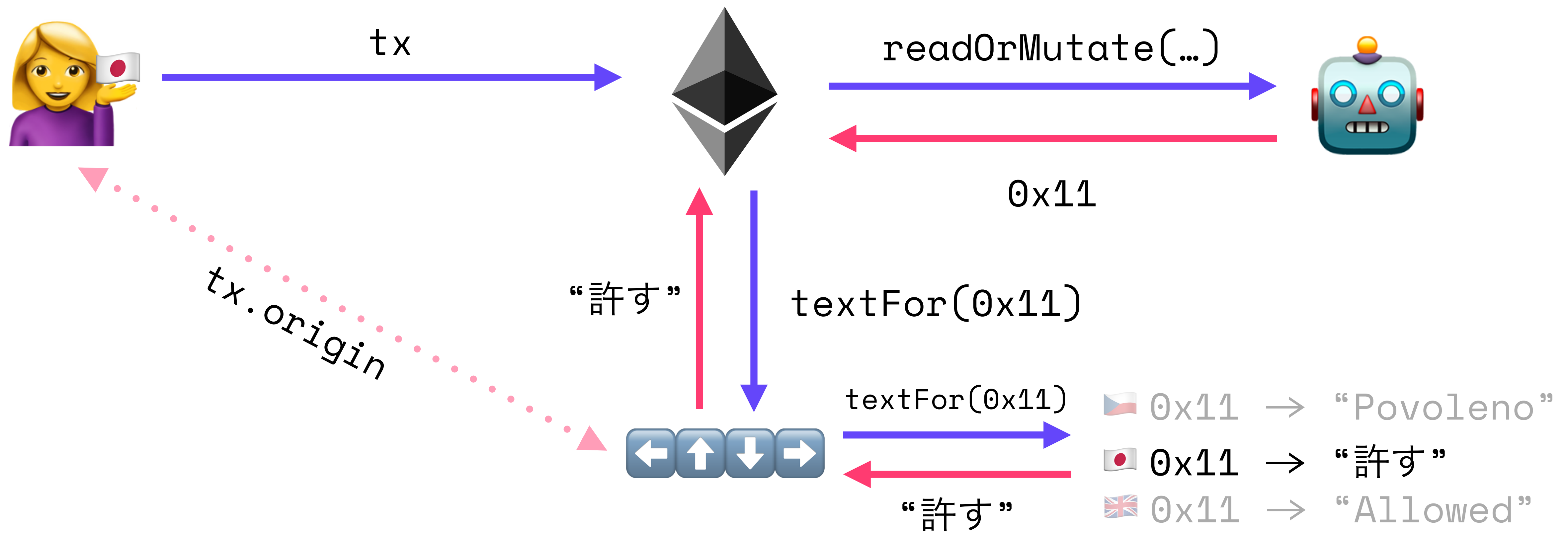
USER FEEDBACK SUCCESS FLOW



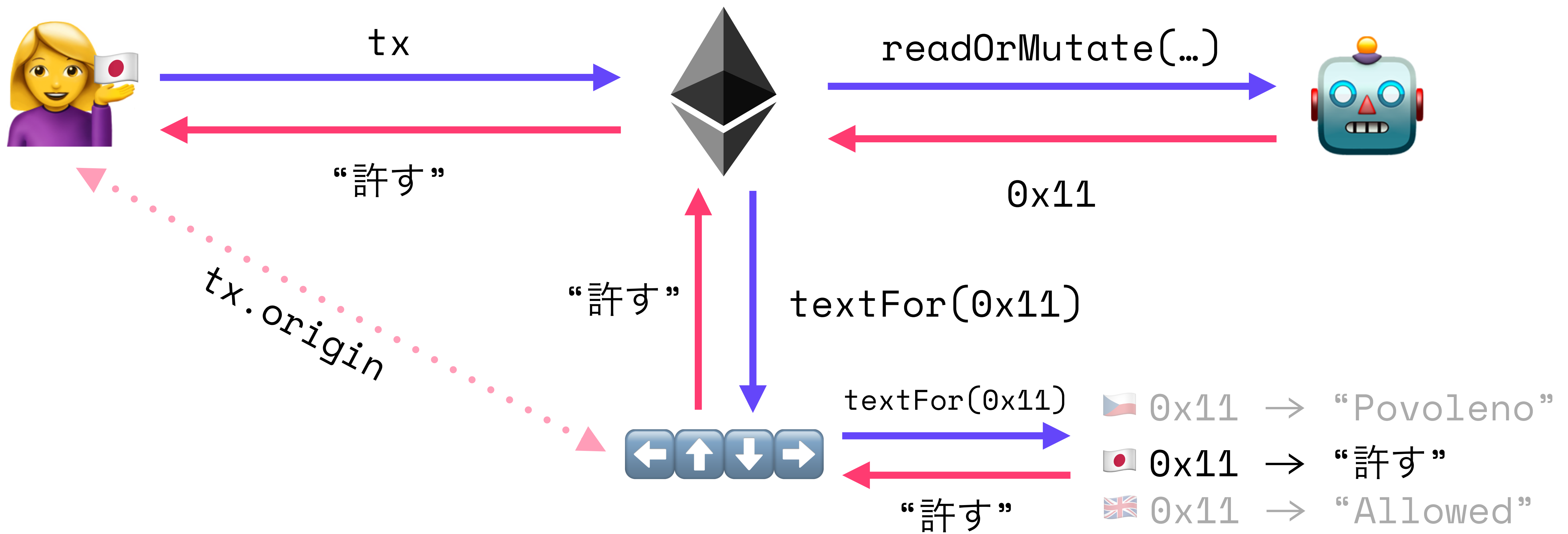
USER FEEDBACK SUCCESS FLOW



USER FEEDBACK SUCCESS FLOW

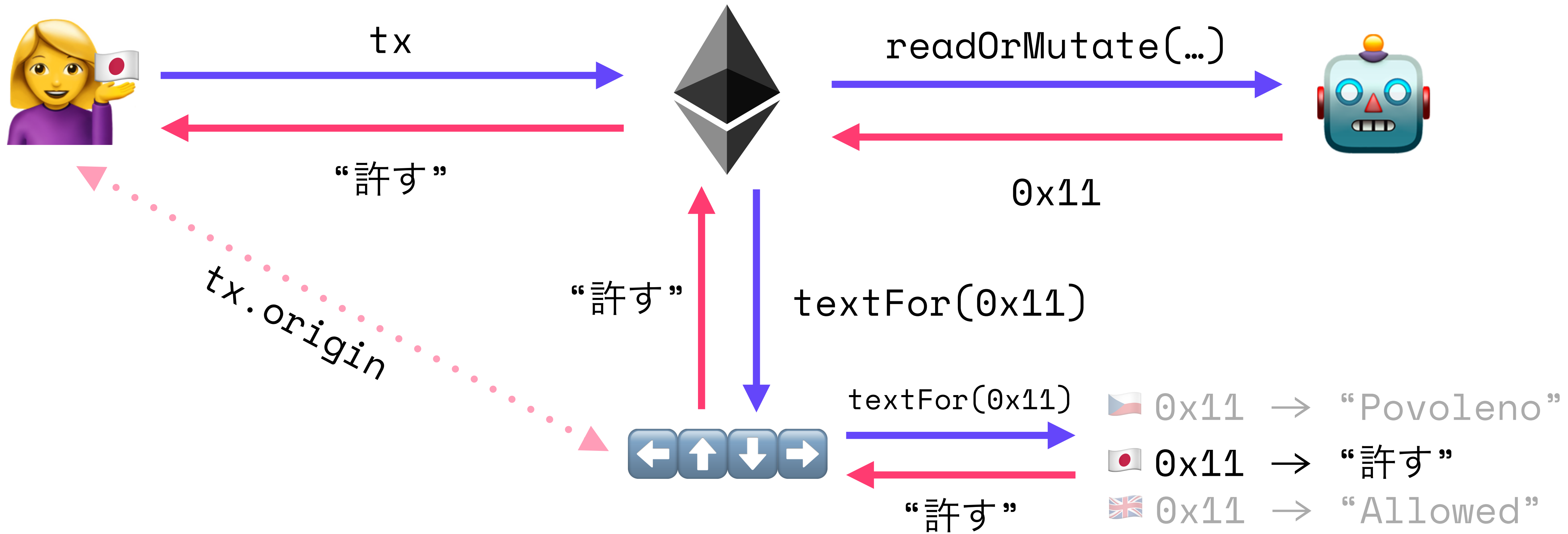


USER FEEDBACK SUCCESS FLOW



USER FEEDBACK SUCCESS FLOW

#3 - 4/25/2018 at 00:05
 AuthorizingContract -0 ETH
CONFIRMED 許す -\$0.00 USD

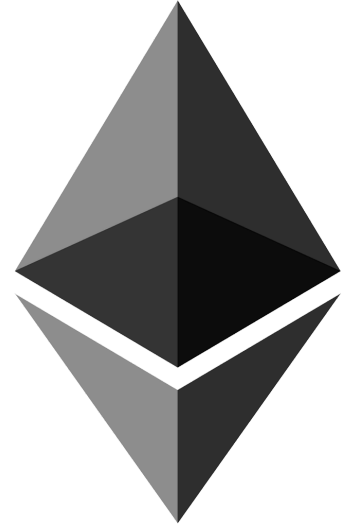


USER FEEDBACK
REVERT FLOW

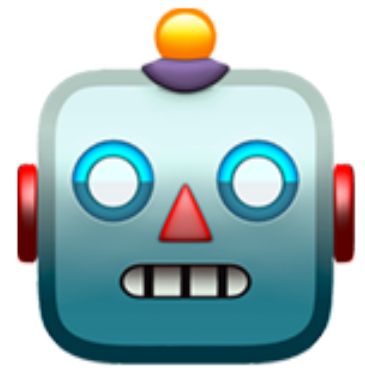
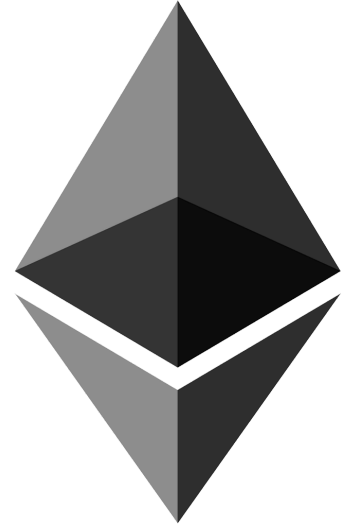
USER FEEDBACK REVERT FLOW



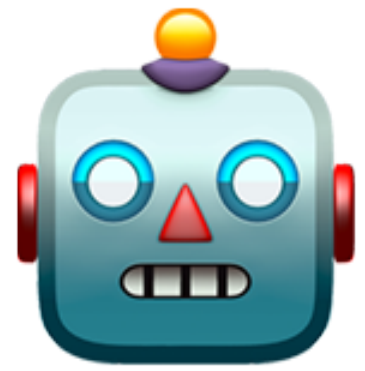
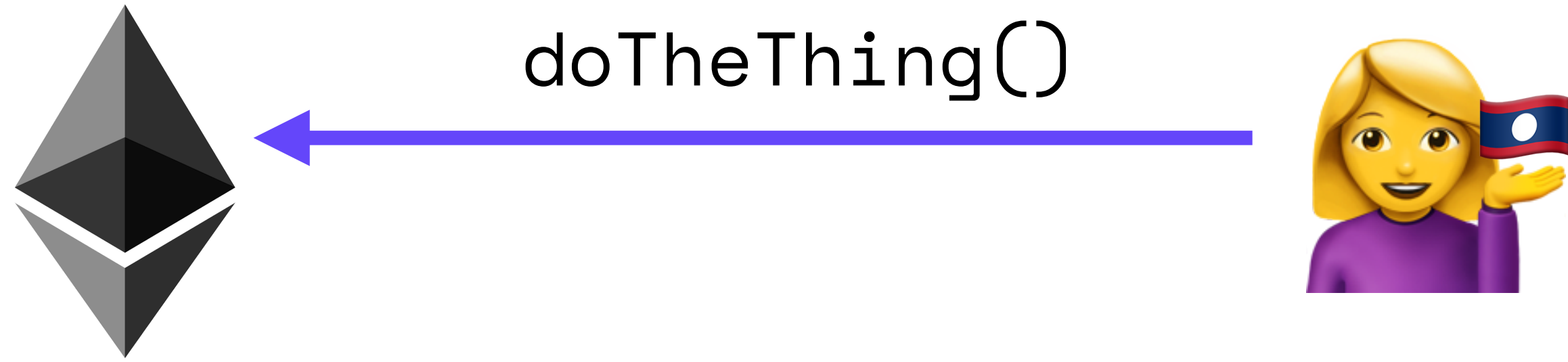
USER FEEDBACK REVERT FLOW



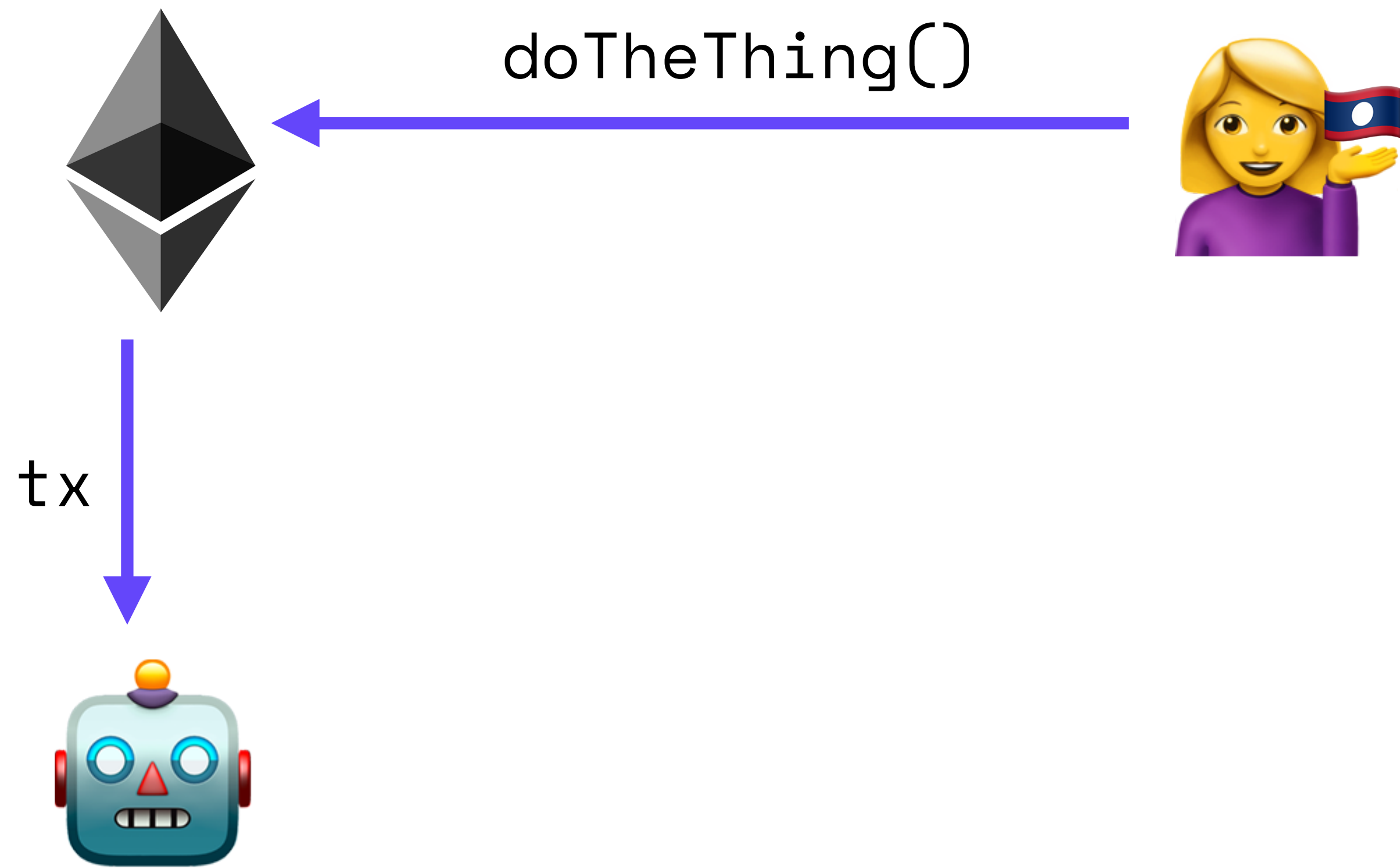
USER FEEDBACK REVERT FLOW



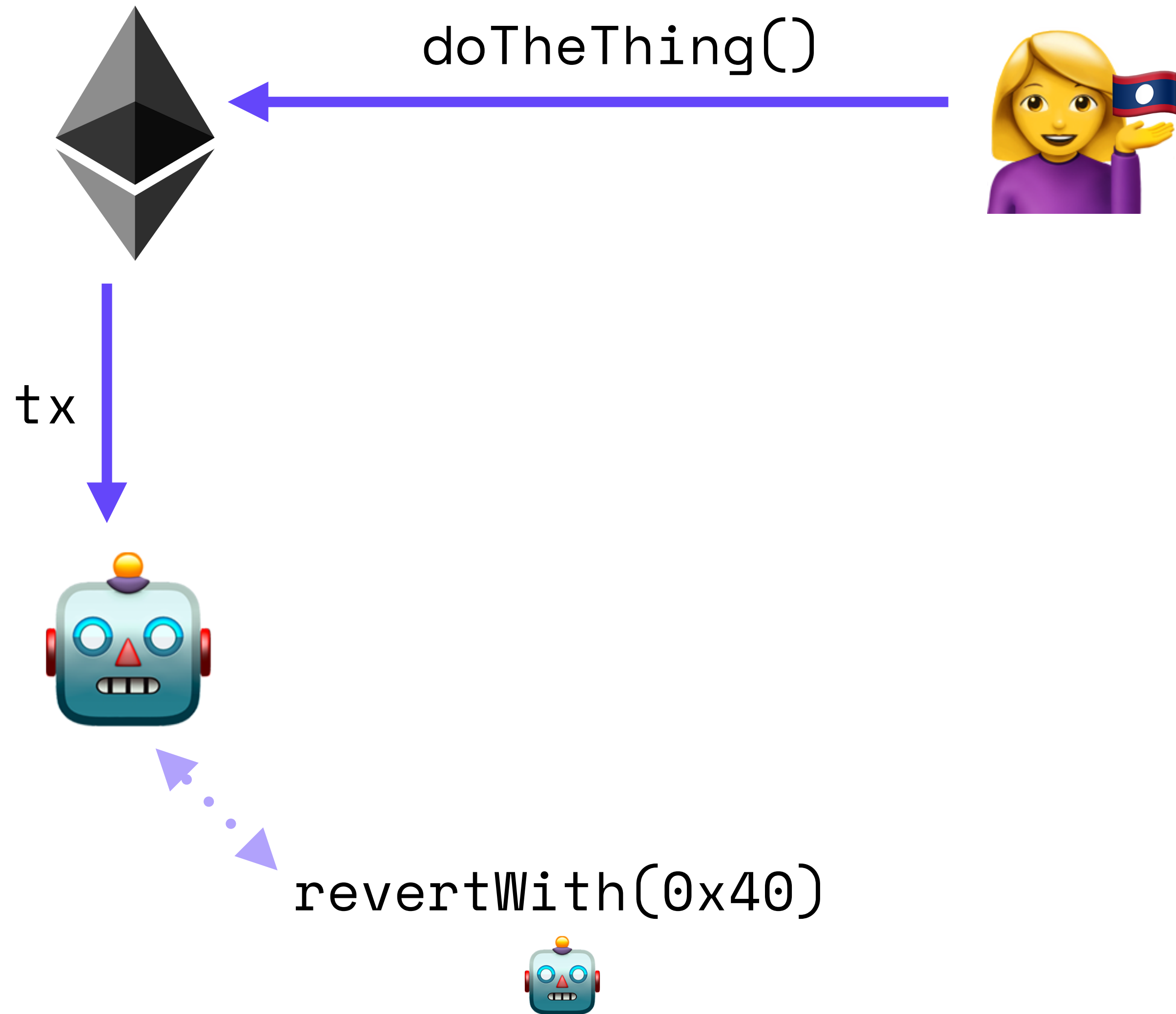
USER FEEDBACK REVERT FLOW



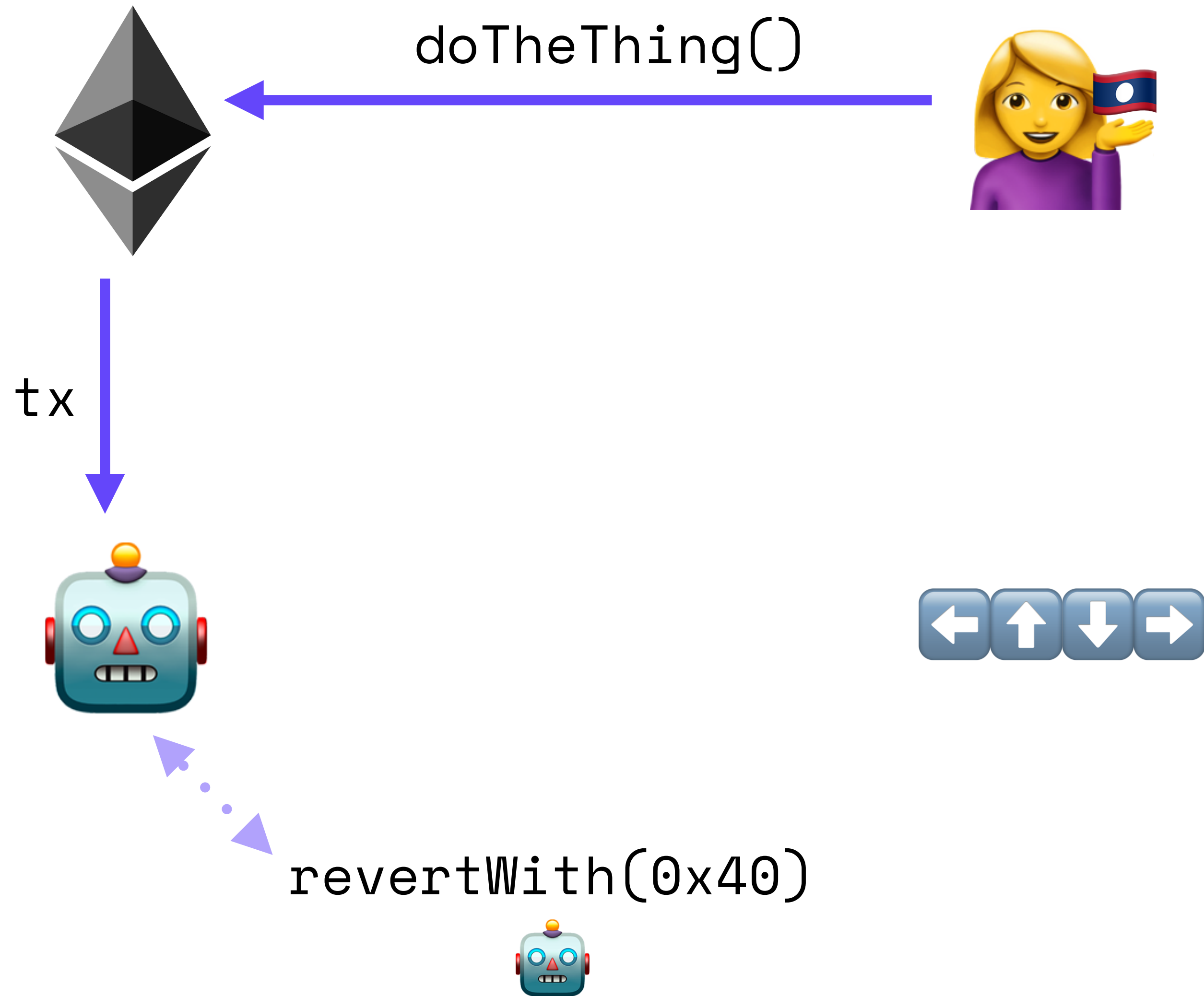
USER FEEDBACK REVERT FLOW



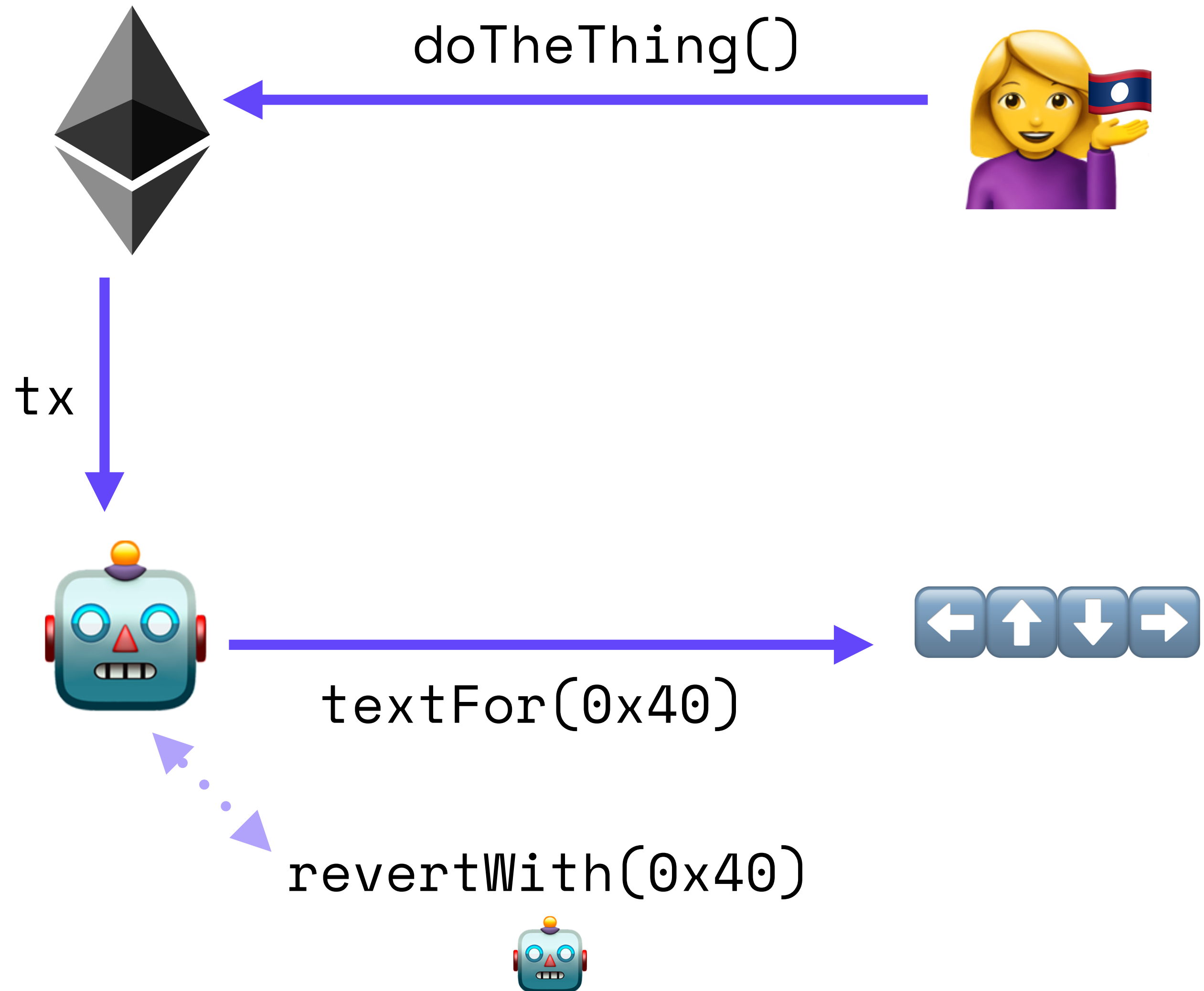
USER FEEDBACK REVERT FLOW



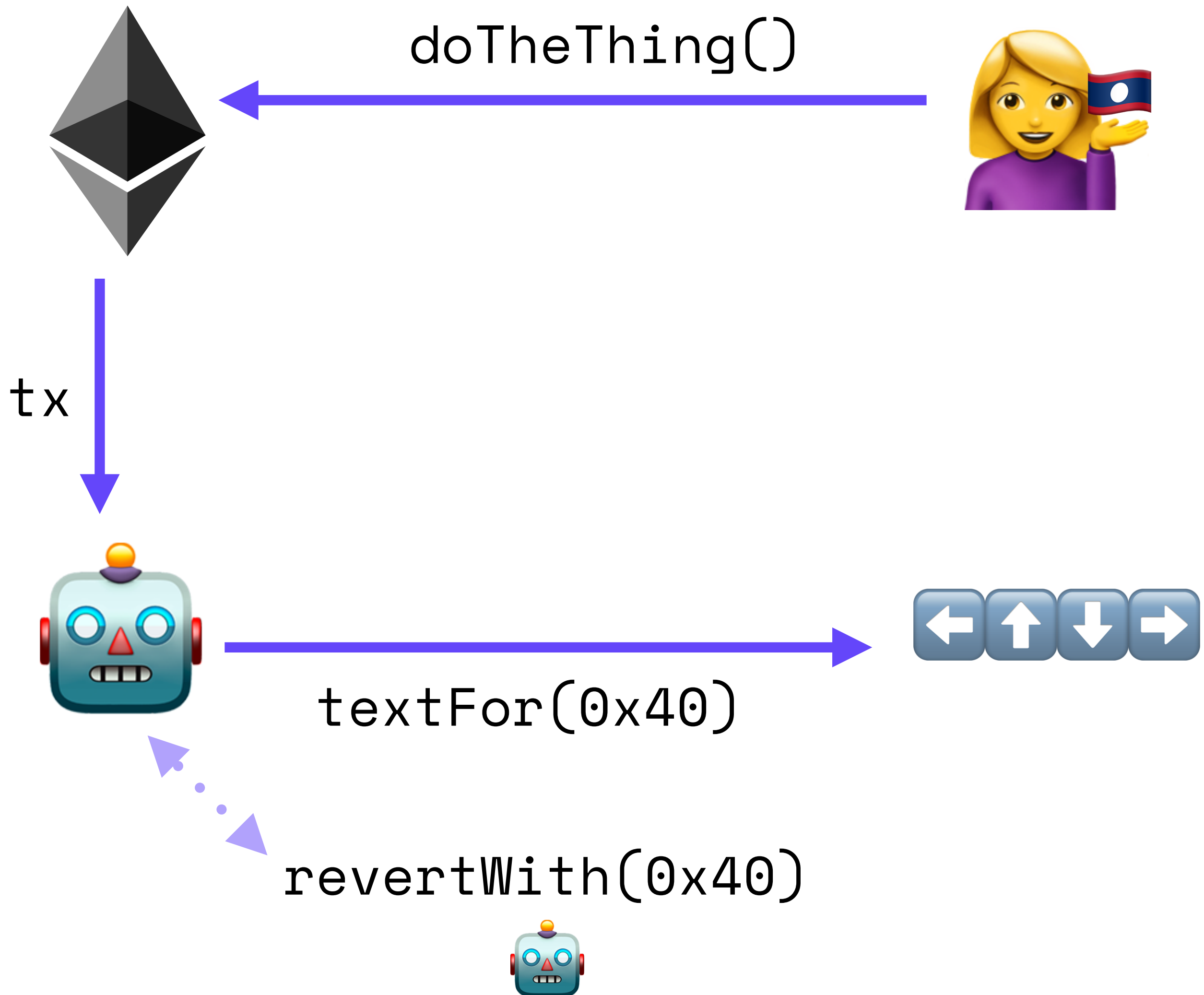
USER FEEDBACK REVERT FLOW



USER FEEDBACK
REVERT FLOW

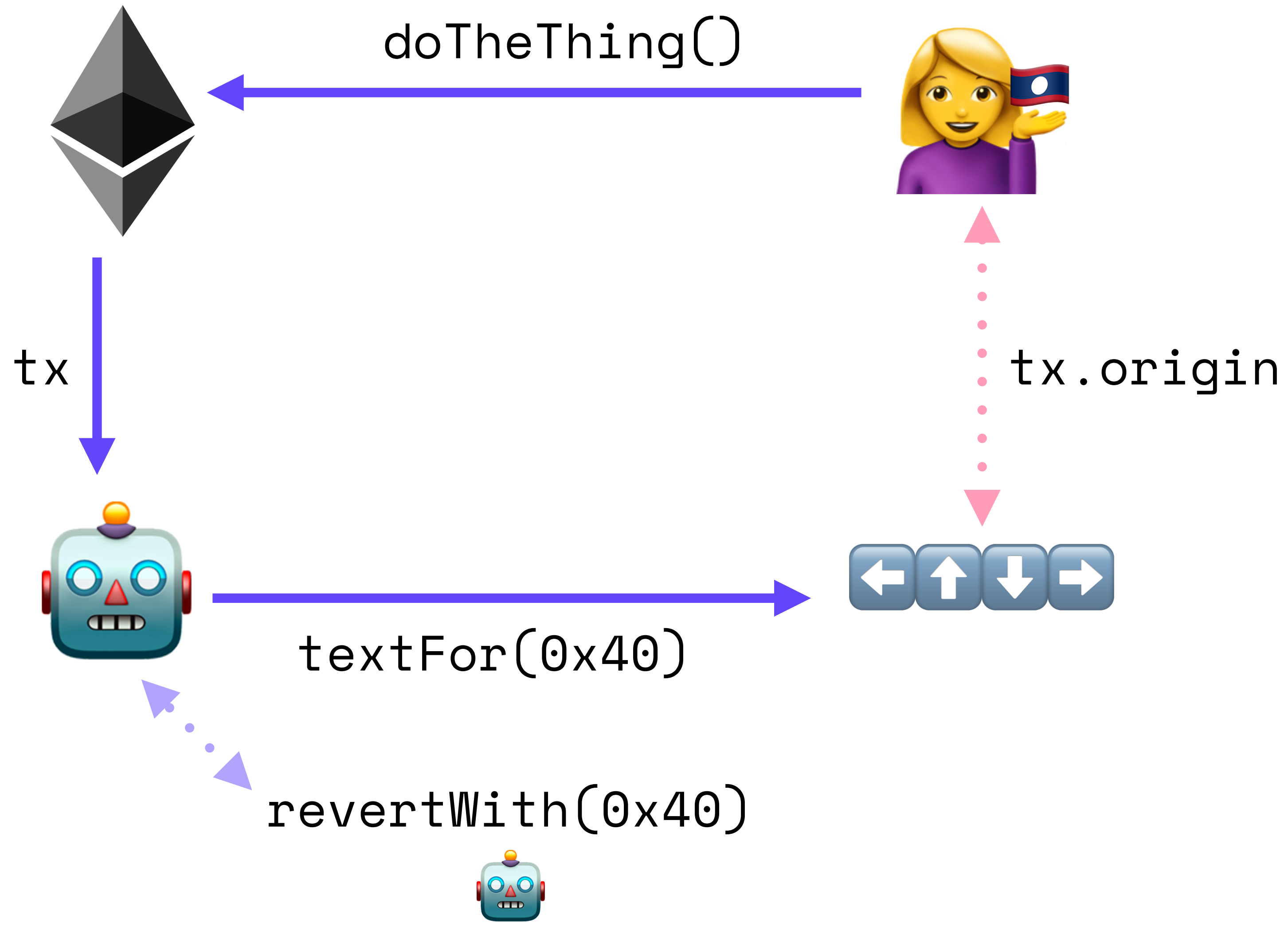


USER FEEDBACK REVERT FLOW



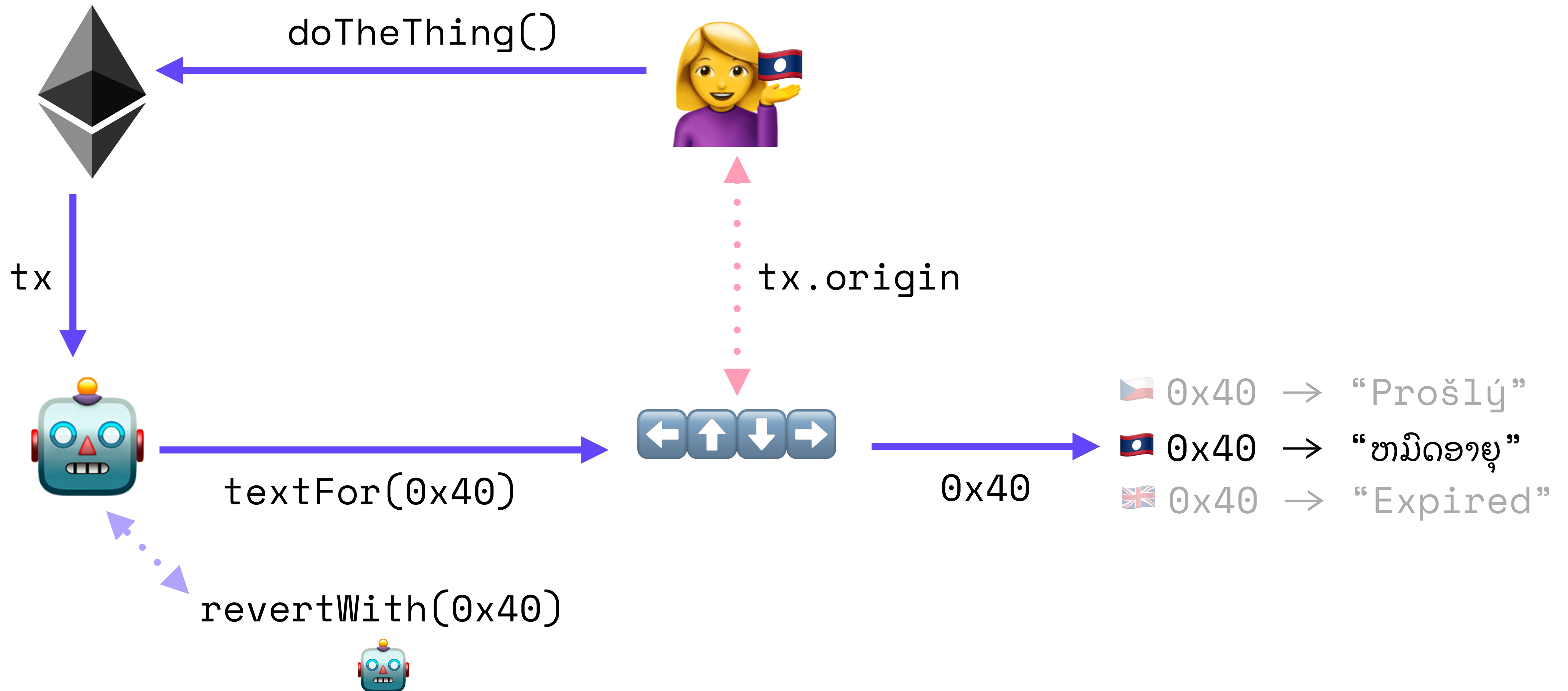
- 🇸🇰 0x40 → “Prošlý”
- 🇹🇼 0x40 → “ຫາມິດອາຍຸ”
- 🇬🇧 0x40 → “Expired”

USER FEEDBACK REVERT FLOW

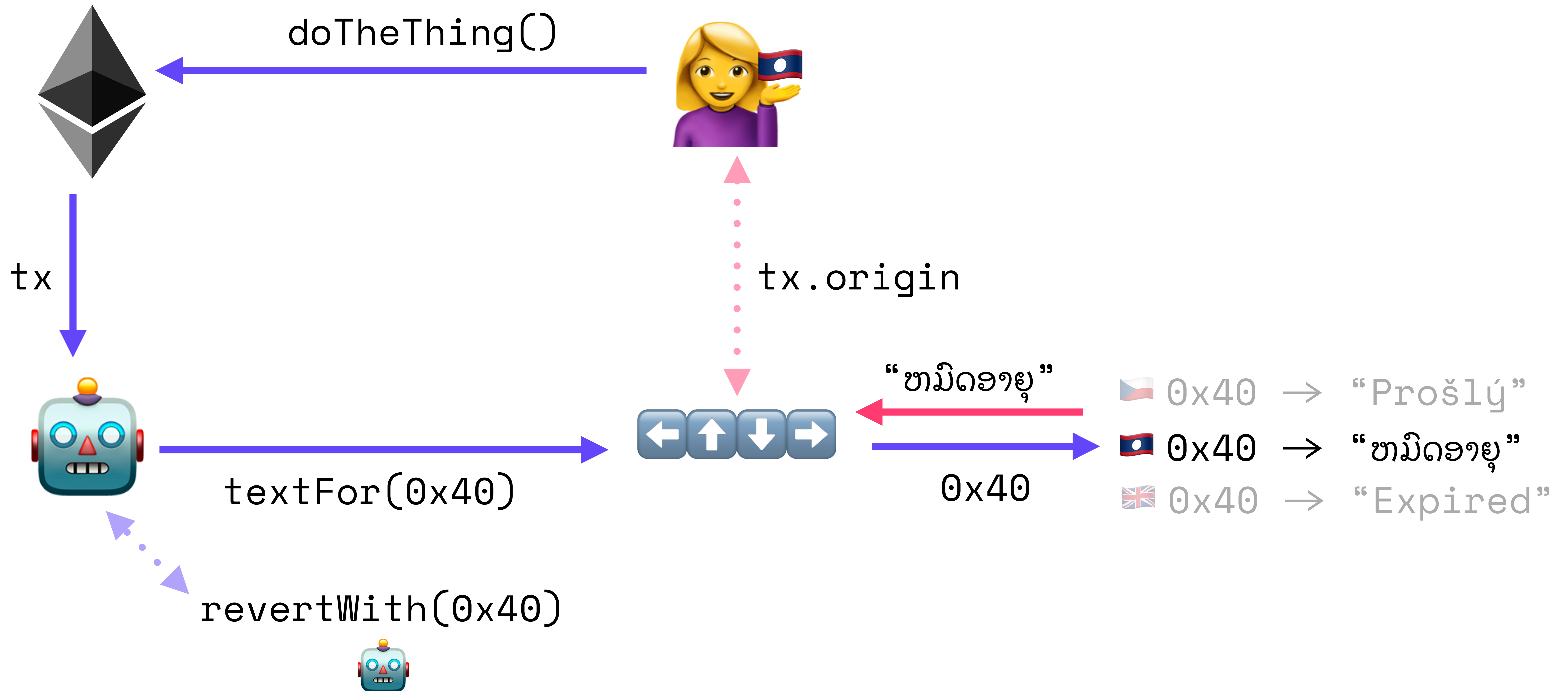


- 🇸🇰 0x40 → “Prošlý”
- 🇹🇼 0x40 → “ຫມົດອາຍຸ”
- 🇬🇧 0x40 → “Expired”

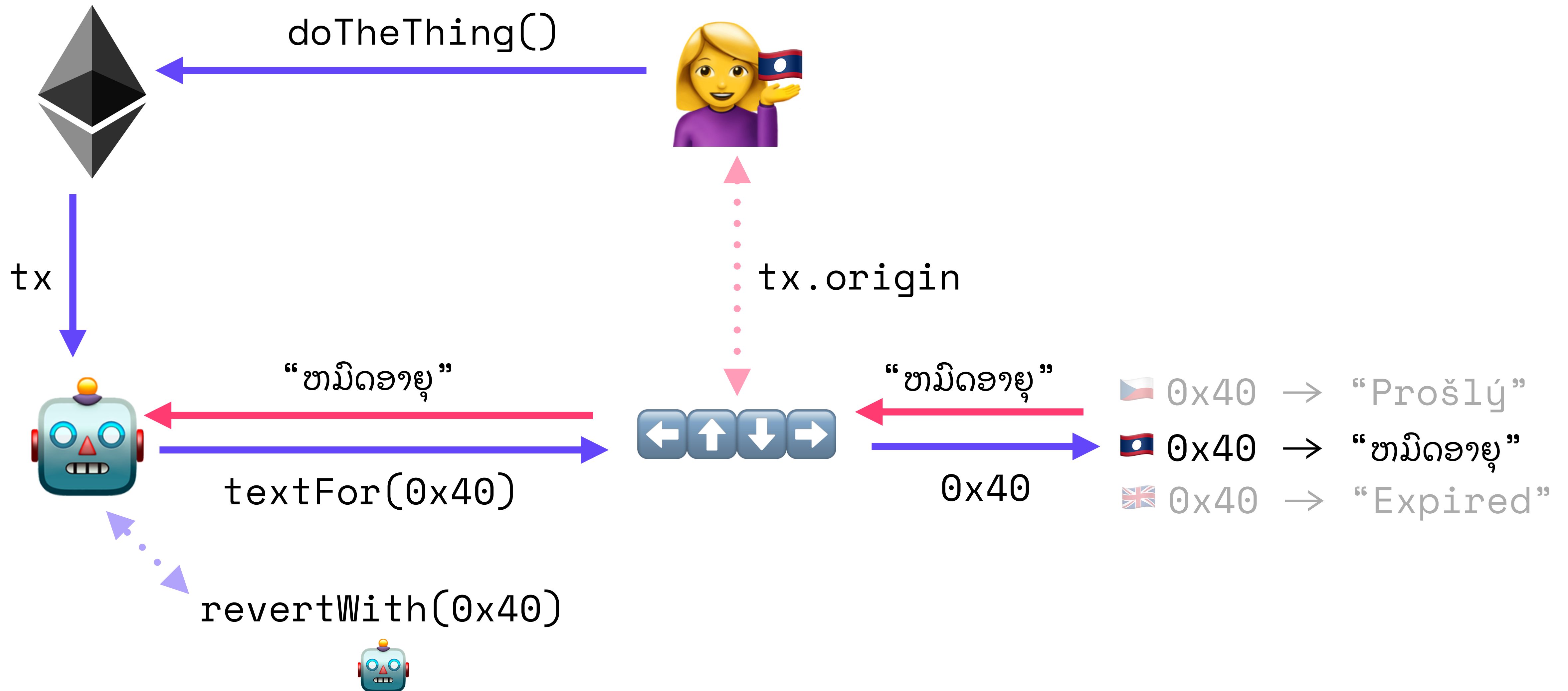
USER FEEDBACK REVERT FLOW



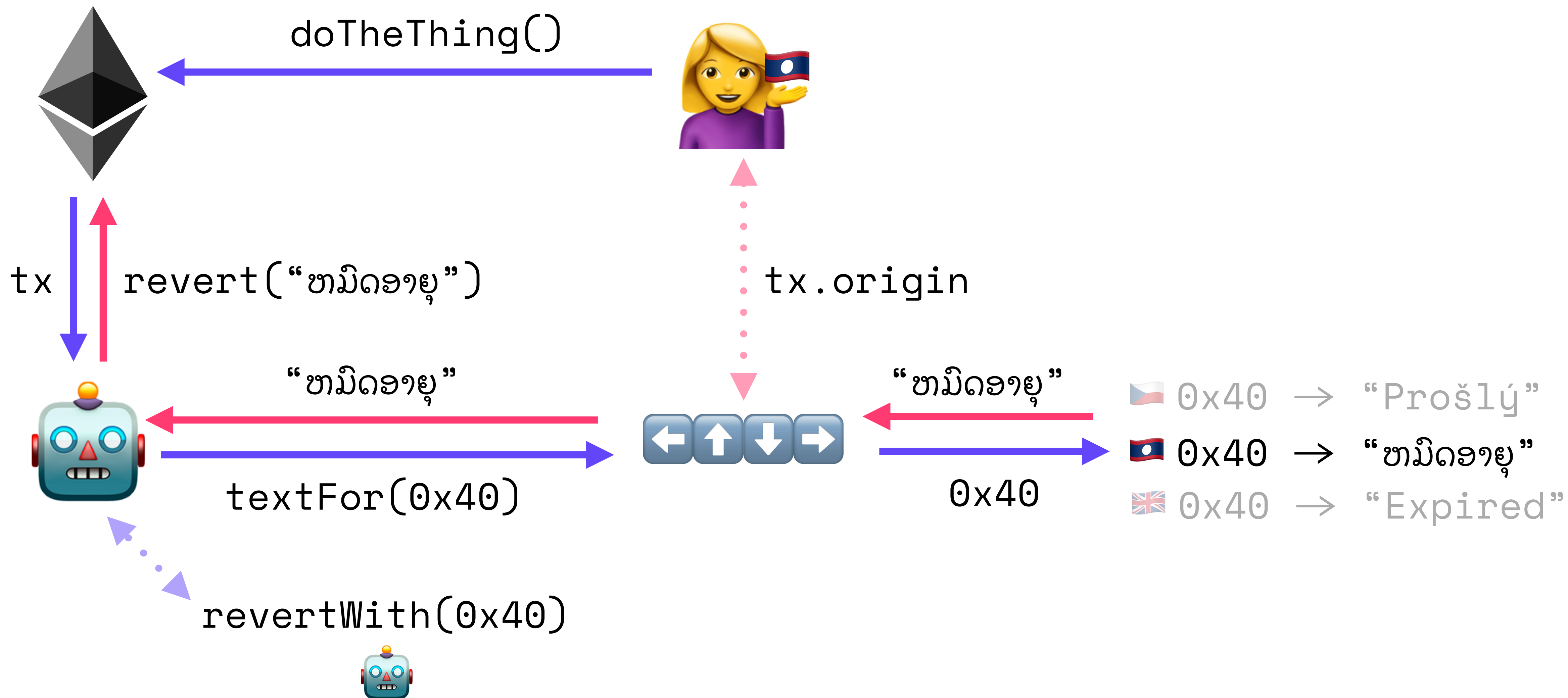
USER FEEDBACK REVERT FLOW



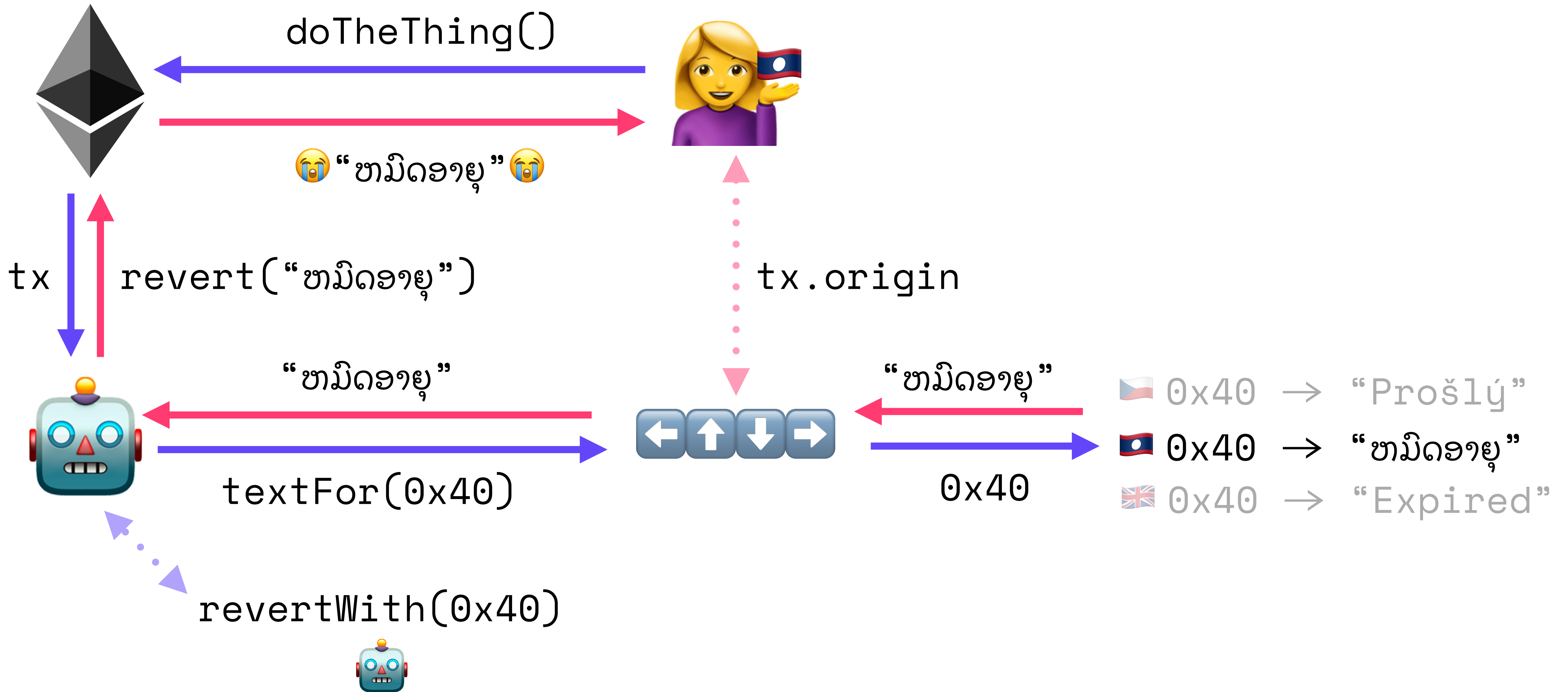
USER FEEDBACK REVERT FLOW



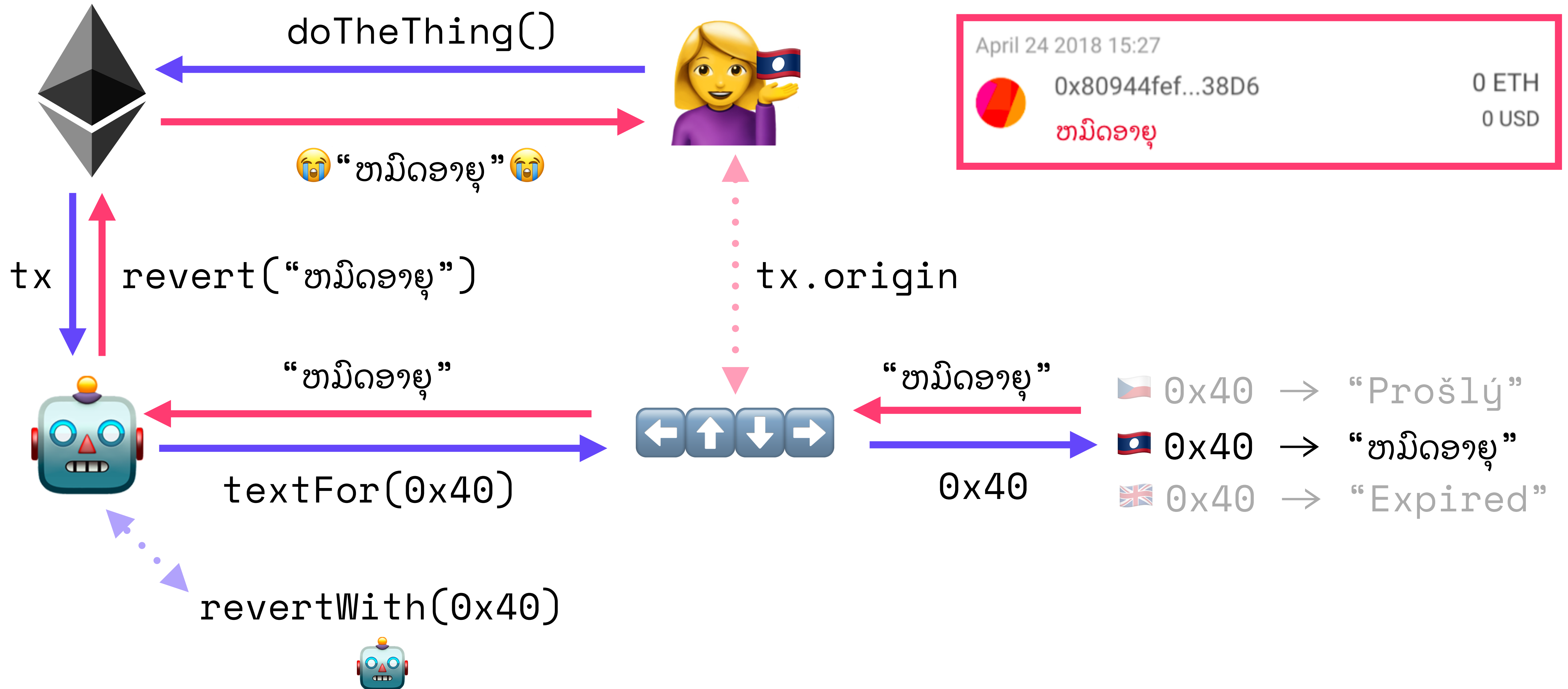
USER FEEDBACK REVERT FLOW



USER FEEDBACK REVERT FLOW



USER FEEDBACK REVERT FLOW



<https://fission.codes>
<https://tools.fission.codes>



THANK YOU, 大阪市



brooklyn@fission.codes
github.com/expede
@expede

