# Types for tokens

## @leithaus - on synereonet.slack.com

Ethereum DevCon II 19 Sept 2016

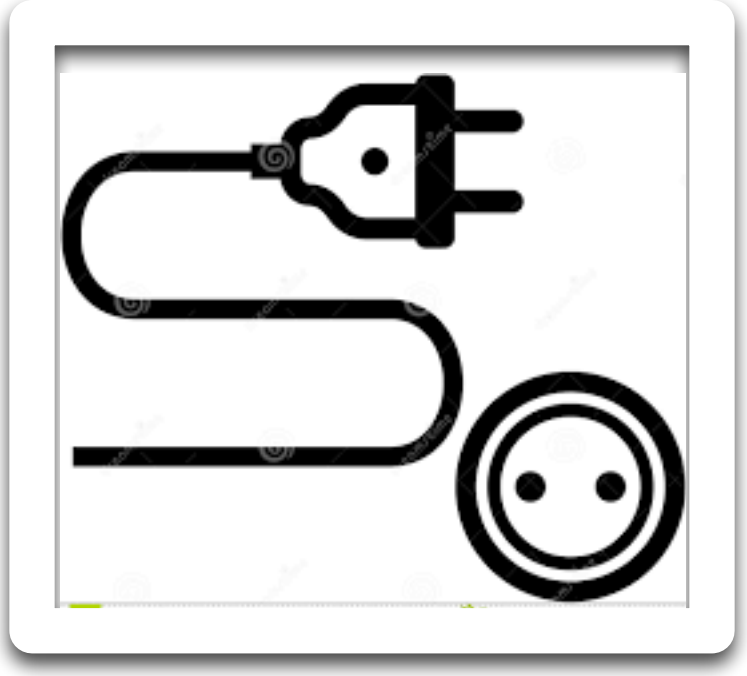
Part I

# What you should get out of this talk if you get nothing else

1. There are new kinds of typing disciplines that make it possible to detect behavioral errors, not just data access errors
2. These disciplines are maturing, finding their way into mainstream programming languages and have a wide range of applications, including security, safety, and liveness
3. This considerably lowers the cost of formal verification and plugs the practice of formal verification directly into common programming practice
***4. We need formal verification for mission-critical systems***
5. The DAO bug is an example of a bug in a mission-critical system
6. Smart / social contracts on a blockchain-based global compute infrastructure constitute a new breed of mission-critical systems

# Types as sockets and plugs

One view
types govern the kind
of data flowing
out of one
program
and
into another



Another view
types govern
when
you can plug
a program
into
an execution
context

What do these new disciplines look like?

# Types as sockets and plugs

f : A => B

f( x ) : B

x : A

traditional

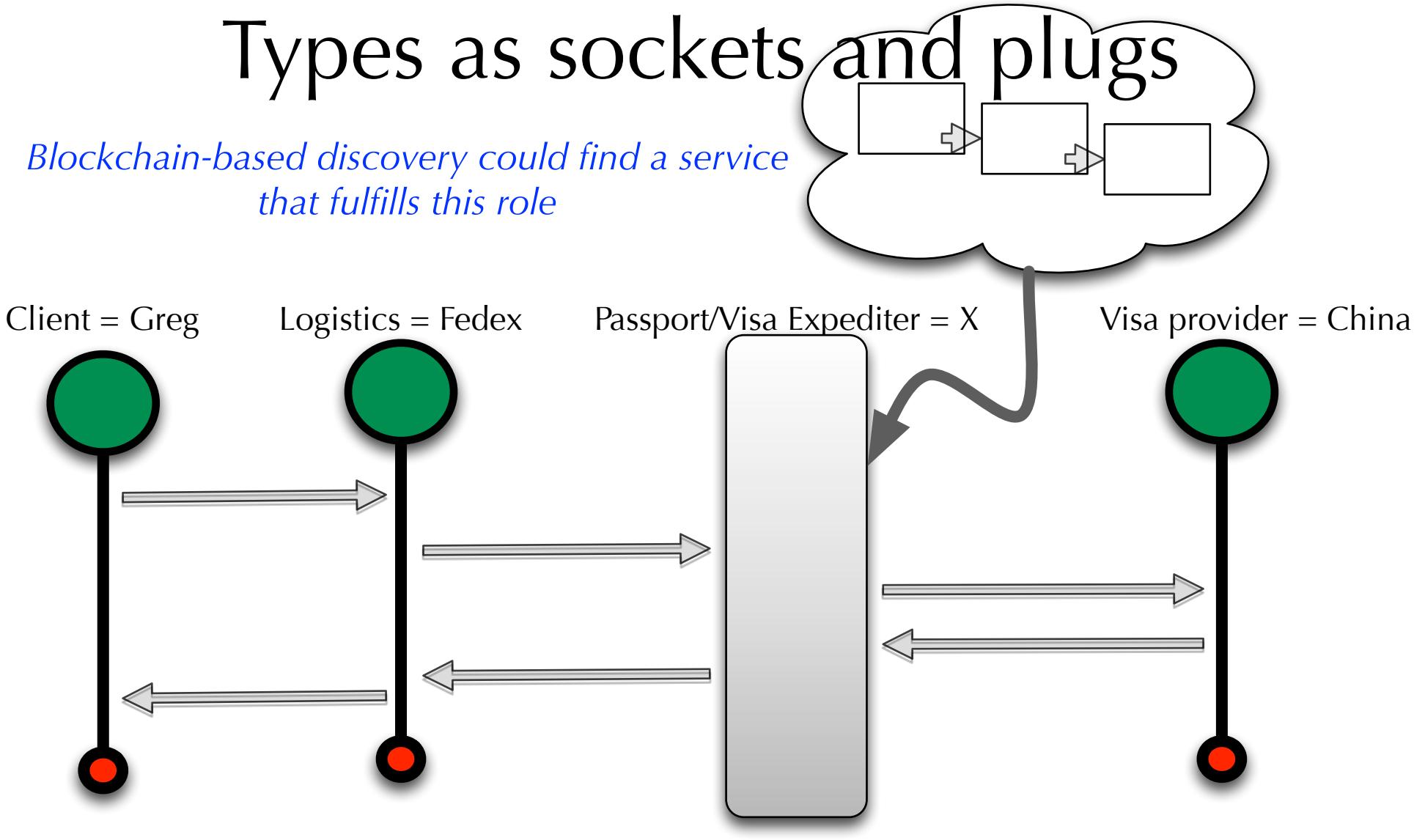
K is a code context
analogous to a motherboard
with slots for additional
components

emerging

whole programs
are plugged into the context
like whole components
plugged into slots
on the motherboard

# Types as sockets and plugs

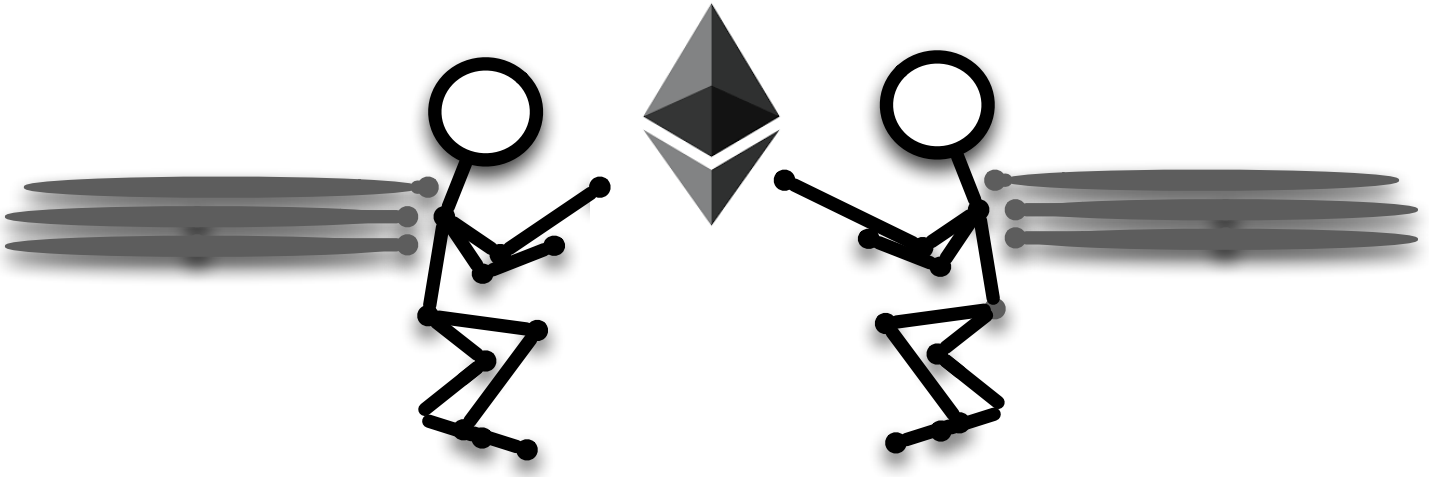
*Blockchain-based discovery could find a service that fulfills this role*




*Business process at Internet-scale was first generation smart / social contract*

# Many kinds of behavioral errors happen at all scales

Example: when the DAO re-entrancy bug is translated into rholang, it becomes a race condition



In a concurrent, distributed system, it matters who gets to non-copyable resources first.
In the case of the DAO, the race was between the update of the state and servicing the next client.

WAIT!

# What is rholang?

Synereo's rholang is a blockchain-based social contracting language
with a built-in
behavioral typing mechanism


smart contracts + fine-grained concurrency = social contracts

***these contracts are social***
***in the sense of Marvin Minsky's***
***Society of Mind***

# Many kinds of behavioral errors happen at all scales

Behavioral types can detect races!
(They can detect a lot of other things, but they can detect races!)
And, races happen at all scales, not just inside one smart contract, but in compositions of contracts and compositions of services that are themselves composed of contracts.

Part II

# Where do these types come from?

1. All programming languages (and a whole lot of other computational phenomena) are generated by monads

L = T + R

T is the grammar for expressions

R are the rewrite rules (think transitions of a VM)

2. Collections (List, Set, Bag, Tree, …) are also monads

C = List or Set or Tree or ...

(SELECT-FROM-WHERE

is derived from the most widely used

monad in programming)

3. Given a distributive law between monads ( T and C ) you can combine them

(T + C)* = Id, T, T2, T3, …, C, C2, C3, …, TC, CT, TCT, CTC, ...

Part II

# Where do these types come from?

(T+C)* -> CT

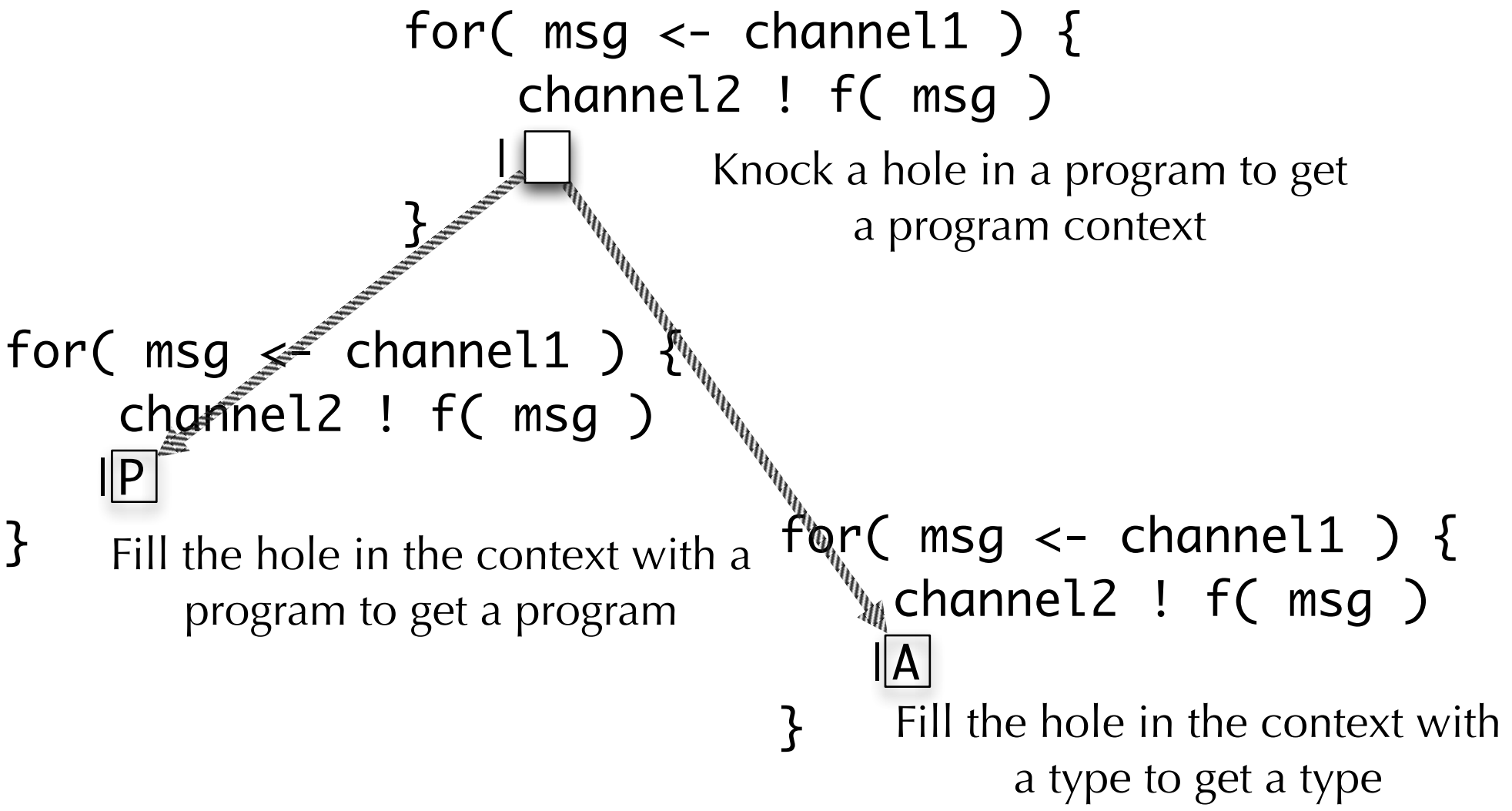
This innocent little fact expresses the semantics of behavioral types

T -> CT, TT -> T, CC -> C (regular monad laws)

TC -> CT (distributive law)

programs with holes in them become collections of allowable programs

# Where do these types come from?



# The reason to express things at this level of generality

1. What we can and have done for rholang with these algorithms is what we can do for solidity
2. What we can and have done for rholang is what we can do for Javascript

Lest this seem terrifically abstract, take a look at K framework which has already provided specs for everything from C/C++ to Javascript to LLVM

we can take K framework specs and add behavioral types to them using our algorithms

# Conclusions

There's a lot more to say about how types can change the blockchain

1. Types can be used to make sharding much more effective
2. Types can be used for service discovery
3. Types can be used to improve Casper

* instead of betting by block, validators bet by proposition on the state of the blockchain - this is one of the improvements Synereo's RChain is developing
* When you type abstract Vlad's algorithm for safety you arrive at a precursor to the bet by proposition we use in RChain

# Conclusions

***Behavioral types provide***
***a way to capture how a Society of Mind***
***(a collection of contracts)***
***expresses a single thought (a type)***

# Types for tokens

@leithaus - on synereonet.slack.com

Ethereum DevCon II 19 Sept 2016