Demo: Light-Weight Programming Language for Blockchain

Junhui Kim (CAU), Joongheon Kim (CAU)
E-mails: jhkim.cau@gmail.com, joongheon@gmail.com

Introduction

• **Bitcoin Script**
→ Difficult to understand and analyze the code what the script language of the Bitcoin has with high-level language style.
→ Since the script language is a sequence of instructions, difficult to understand the flow of codes at a glance.
→ Tricky to implement complicated logics as instructions.
→ Difficult to understand what kind of actions to perform.
→ Difficulty to program considering the status of stack.

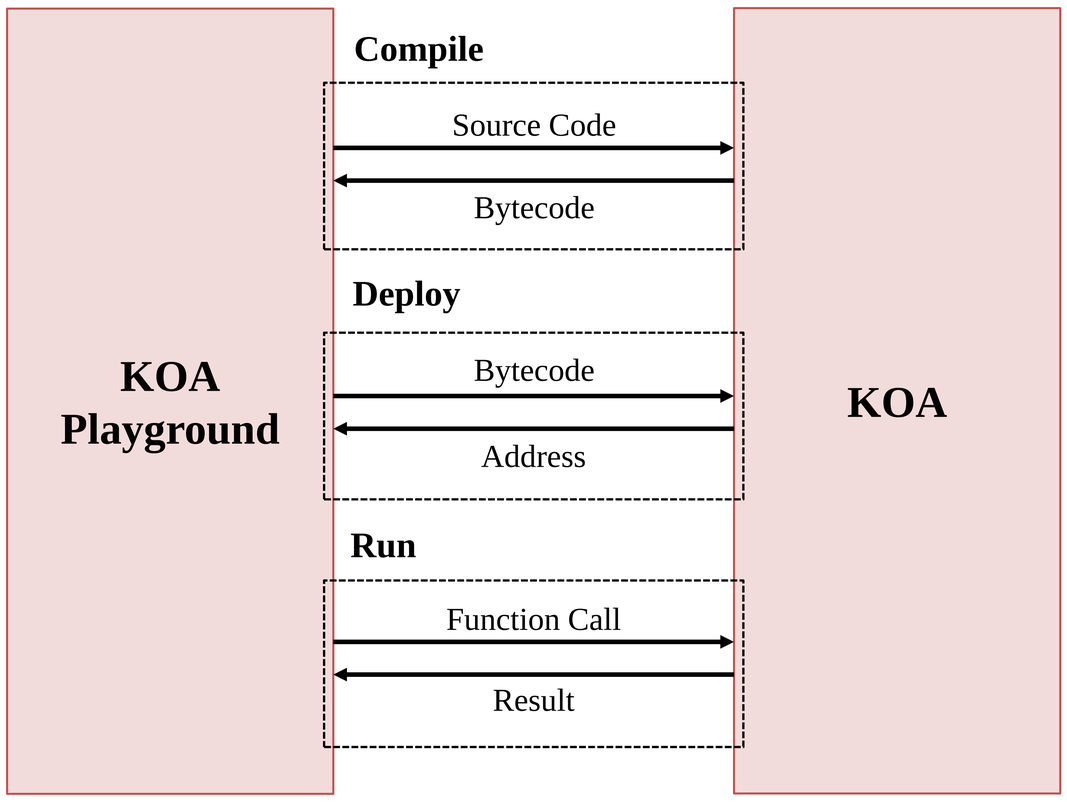
• **Solidity**
→ The gas makes solidity code more complicated and hinder usability.
→ The state of Ethereum makes the static analysis of solidity code difficult.
→ Since the result of executing function differs according to state, it is difficult to predict gas consumption.

A new programming language which solves difficulties of understanding and analyzes the code

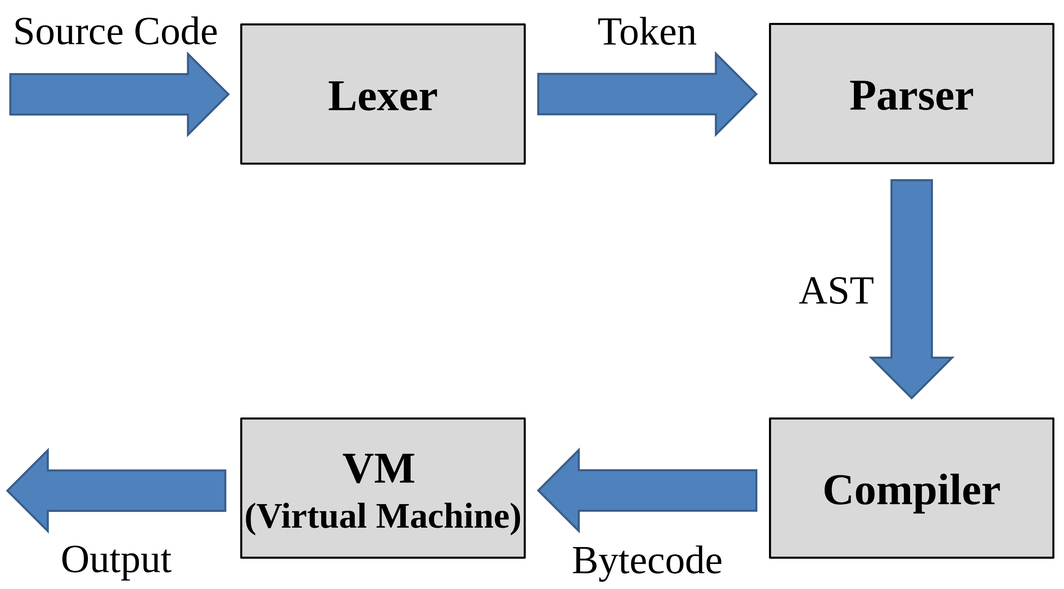
• **Koa**
→ Because it is stateless language, it can be statically analyzed..
→ Since there are not any branches of code execution depending on state nor loop, static analysis for the code is easy.
→ It can prevent the infinite loop attack.
→ It can predict the costs associated with running the smart contract.
→ It helps programmers to implement the blockchain quickly and easily.

System Overview

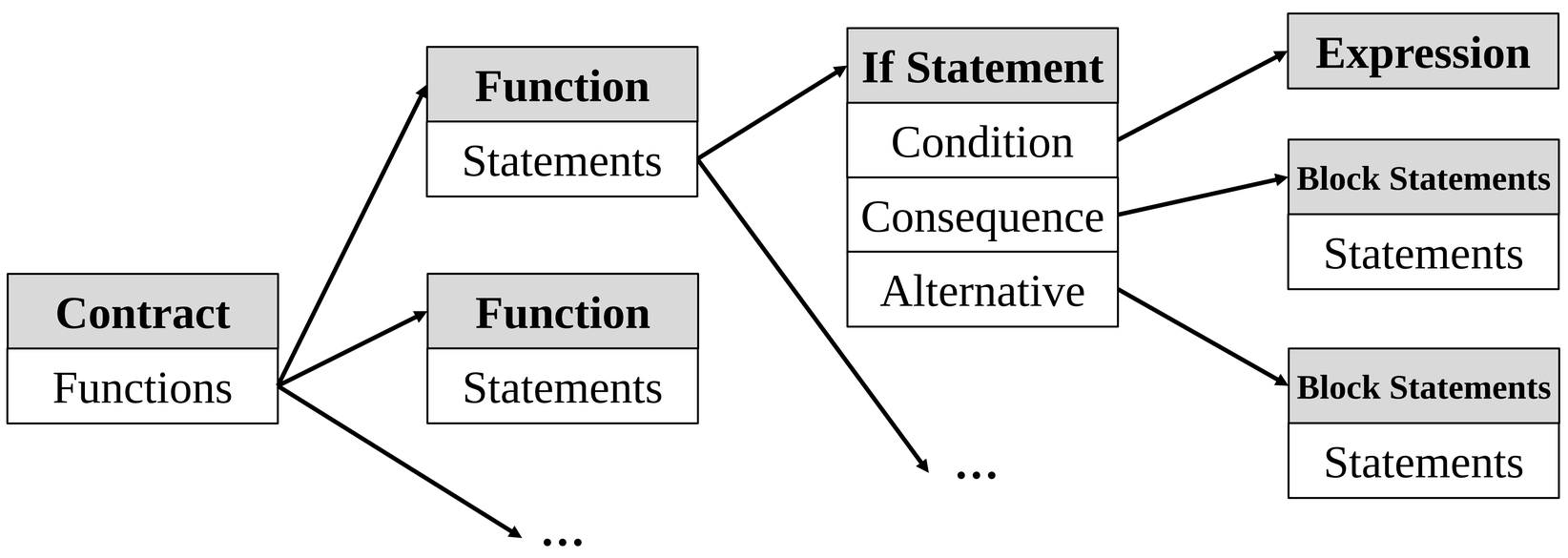
KOA System



KOA Architecture



Abstract Syntax Tree



Code Example

```
contract {
    func Sig(sig string) {
        string pubkey =
"fvfidBGruUYCmTw7CusaCOQbBuZBiYdu"

        if checkSig(pubkey, sig) {
            return true
        }
        return false
    }
}
```

ACM International Conference on Mobile Systems, Applications, and Services (June 18, 2019)
Seoul, South Korea