

# 一个自我进化的通用智能合约 开发平台 (A Self Evolving Universal Smart Contract Platform)

## 简介

一个好的区块链的智能合约平台，如何吸引更多的开发人员在上面开发DApp，是这个区块链智能合约平台成功的关键。而要吸引更多的开发人员，一个好的智能合约开发语言是其中的关键。

一个好的智能合约开发语言，不仅能够提高开发的效率，节省开发者宝贵的开发时间和精力，也能减少智能合约的开发过程中出错的几率，提高智能合约代码的健壮性和可维护性。

就像苹果系统和安卓系统一样，一个开放的智能合约开发平台的前景如何，取决于有多少开发者在上面开发软件。苹果系统上可以用Objective C和Swift来开发应用软件，并且有强大的Xcode 开发工具提供支持。安卓系统上首选Java作为开发语言，同样可以用Eclipse和Android Studio这些强大的IDE工具作为开发工具。

但是相对于现在的兴旺发达的通用软件开发行业，区块链的世界则是显得要落后许多，落后到甚至智能合约的调试都只能通过输出日志来进

行。当然，这也从一个方面说明了现在的区块链的发展真的还属于非常早期的阶段。但是，另一个方面，也说明了区块链还有很大的发展潜力。

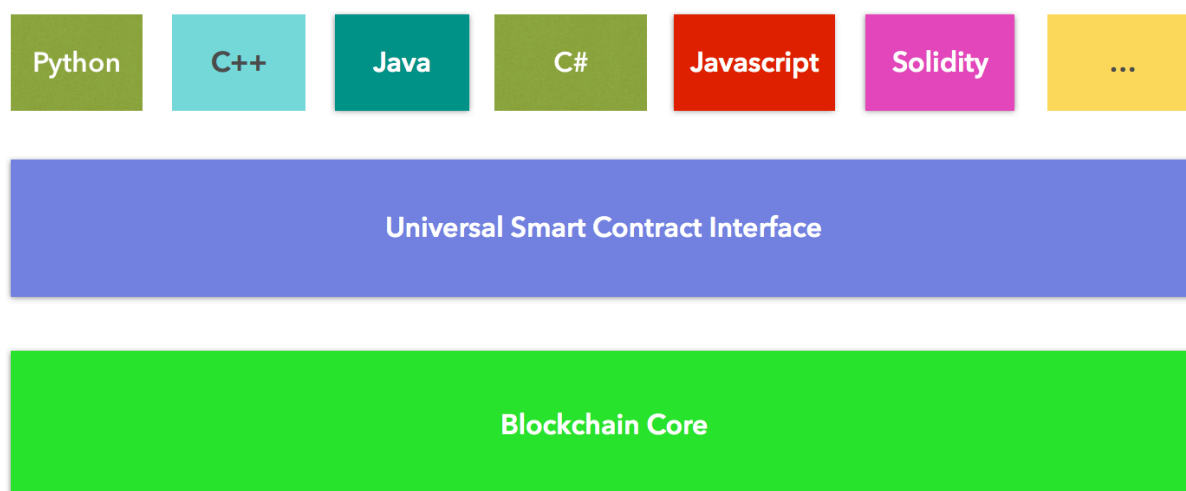
这其中的原因，很大程度上是因为区块链这种金融软件对安全需求非常之高，客观上限制了智能合约的实现不能太复杂。越复杂的智能合约意味着越高的开发难度，随之而来的是更高的安全风险。

如何在保证智能合约安全性的情况下，提高智能合约的易用性和开发效率，是当前区块链世界必须解决的问题。

PyEos正是为了解决上面的问题而生的，PyEos将Python作为它的主要的智能合约开发语言。Python的强大不用多说了，稍微有点编程知识的人应该都听说甚至用过Python，Python因其简单易用，开发效率高，并且能快速上手的特点，得到了广大开发者的喜爱。所以能用Python来开发智能合约，是所有智能合约开发者之福了。

PyEos除了主要支持Python作了主要的智能合约开发语言之外，并没有抛弃其它语言。PyEos保留了Eos上的WASM智能合约，并且考虑到以太坊上的巨大的智能合约开发群体，同时也支持Solidity智能合约语言。并且PyEos已经在架构上作了支持更多语言的准备。希望借此能吸引更多的开发者开发DApp。

## 项目架构



PyEos采用了类似Android系统的架构，将大多数的智能合约的运行从区块链代码中独立出来，单独运行于一个或几个运行环境高度受限的进程之中。区块链主程序和智能合约代码之间通过IPC的方式进行通信，主程序能实时对智能合约进程进行监控。智能合约进程出了问题，也不会影响到主程序和整个网络的。

并且，由于链上的数据是主程序和智能合约进程通过Memory Map技术共享的，再加上并行等优化手段，可以使进程间通信造成的性能损失降到最低。用很小的性能代价，来换取更高的安全性，对于视安全为生命的区块链项目来说，是非常值得的。

同时，PyEos也支持以RPC的方式来进行区块链主程序和智能合约进行通信，一些需要大量的计算资源的Transaction可以通过RPC的通信机制运行于其它主机之上，从而使PyEos的超级结点既满足存储中心的需求，也满足计算中心的需求。这为将类似AI这样的程序运行于区块链之上提供了可能。

当然，对于一些经常要用到的安全的智能合约程序，则可以跑在主程序之中。

在该架构下，智能合约的开发也不再受限于某种语言。从理论上讲，只要是实现了智能合约接口的编程语言都可以用来实现智能合约的开发。

通过以上的设计，在保证保证智能合约的安全性和性能的前提下，解决智能合约的易用性和开发效率

## 项目功能简介

## 一、自我进化

**PyEos**已经可以做到通过链上更新**WebAssemble**, **Micropython**以及**EVM**虚拟机以及更新**Native**智能合约。将来会对这个功能进行进一步的扩展。**PyEos**上的所有功能都将会是一个一个**Component**，可以通过链上被自由替换。这其中包括**DPOS**的共识算法，以及通过主链发起一条并行链或者侧链的功能。打个比喻来说，**PyEos**将实现类似在飞机飞行的过程中更换轮子，甚至更换引擎的功能。

## 二、Native 智能合约

**PyEos**已经做到可以将一个**WASM**智能合约编译成**Native Code**并通过链上进行发布和调用。这将智能合约的运行速度提高了**3到8倍**。其调用过程和**WASM**智能合约无异。智能合约的调用者感觉不到这是一个**WASM**智能合约还是**Native**智能合约。

## 三、账户信用管理系统

一个发展良好的智能合约开发平台，需要社区成员的共同来维护。如何对区块链上的账户进行管理，是值得每一个区块链项目深入探寻的问题。本项目试图通过一个智能合约来管理账户的信用，建立账户的信用等级，来维护和促进整个区块链生态的发展。

## 四、支持部署和运行以太坊智能合约

以太坊的**EVM**虚拟机，是一个比较成熟的虚拟机了，是目前少数几个已经正式商用的用于智能合约运行的虚拟机。而面向**EVM**而开发的**Solidity**智能合约语言，有着大量的开发资源，得到了广大的智能合约开发

者的支持。所以，虽然EVM的运行比较慢，尽管Solidity语言难以使用，但是还是很有必要支持的。

## 五、支持在同一账户之上部署多份智能合约代码

将智能合约代码分割成不同的模块是很有必要的，其优势很明显：

1. 只加载需要加载的代码，减少代码的加载时间
2. 方便管理和升级

## 六、支持跨帐户之间的智能合约的相互调用

链上的存储空间是很宝贵的，而不同的智能合约之间实际上有相当多的重复代码，这些重复代码可以说是对链上的存储空间的巨大浪费。有了这个功能，智能合约就可以通过调用其它智能合约的代码来节省存储空间。

## 七、支持代码租赁和出售

有了第四个功能的支持，开发者通过链上代码租赁和出售的方式来获取收益成了一种可能。通过一个CodeStore（代码商店）智能合约，在PyEos底层的支持下，可以实现类似苹果的AppStore的功能。开发者可以将代码上传到CodeStore智能合约，等审核通过后即可在上面通过集体竞价或者直接买卖的方式出租或者出售自己的代码供别人调用。从而使开发者从开发代码的过程中直接收益，在节省可观的链上资源的同时，也可以大大提高开发者开发通用DApp的积极性，促进整个区块链生态的发展。

## 八、多编程语言支持

PyEos除了支持C++作为智能合约开发语言，目前还支持Python, Solidity作为智能合约开发语言。其它像Java、C#、Javascript、Go这样的

语言也在计划之中。支持更多的语言意味着能够吸引到更多的开发者。而一个智能合约平台的生态直接取决于有多少开发者在这个平台上开发 DApp。所以，在保证安全的前提下，支持更多的编程语言是很有必要的。

## 九、支持链上智能合约代码的源码级调试

现在大多数的智能合约平台，是不支持源代码的断点调试的，目前 Eos 也不支持。这不能不说是个缺憾。由于 PyEos 支持已经发展了二十几年的已经发展相当成熟的 Python 作为主要的智能合约语言，使智能合约的调试变得非常简单容易。开发者可以很方便的用像 Eclipse, Visual Studio Code 这些强大的 IDE 来进行智能合约代码的调试，如下图所示：

