

Welcome Neoxplorers!

Enterprise Distributed Application Development using the NEO Blockchain

Michael Herman

Independent Blockchain Architect and Developer

NEO C# Developers Center of Excellence

neotoronto@outlook.com

1. Ulusal BLOKZİNCİR ÇALIŞTAYI

ANA SPONSOR

TÜRKİYE CUMHURİYET MERKEZ BANKASI

Misafir Konuşmacılar

Peter Gaži
Research Fellow, co-author of the Ouroboros Praes protocol - the next version of the Cardano system
IOHK research iohk.io

Malcolm Lerider
Senior R&D Manager
NEO Smart Economy neo.org

Konular >

- Bitcoin ve Diğer Kripto Paralar
- Merkezi Olmayan Sistemlerde Bizans Hata Toleransı
- Blokzincir Teknolojisinin Gerçek Hayattaki Muhtemel Uygulamaları
- Blokzincir Ekosistemi: Finans, Enerji, Tedarik Zinciri, Depolama, Hesaplama, Haberleşme
- Kripto Paraların Güvenlik ve Mahremiyeti
- Akıllı Kontratlar
- Kitle fonlama, ICO'lar
- Blokzincir Teknolojisinin Devlet Kurumlarına Uyarlaması
- Sahetcilik ve finansal kriz önleme yöntemleri
- Kripto paralarda ekonomi ve oyun teorisi analizi
- Emeğin İspati (PoW), Hisse senin İspati (PoS)
- Kullanılabilirlik ve Son Kullanıcı Senaryoları
- Kripto Paraların Adaptasyon ve Uygun Süreci
- Regülasyonlar ve Yasal Düzenlemeler
- Merkezi Olmayan Yapılara Hukuki, Etik ve Sosyal Bir Yaklaşım

Özet Çağrısı >

Son Başvuru Tarihi 11 Mart 2018
Sonuçların Açıklanması 19 Mart 2018

BİLGEM | **BLOKZİNCİR** Araştırma Laboratuvarı

NEO | **CARDANO**

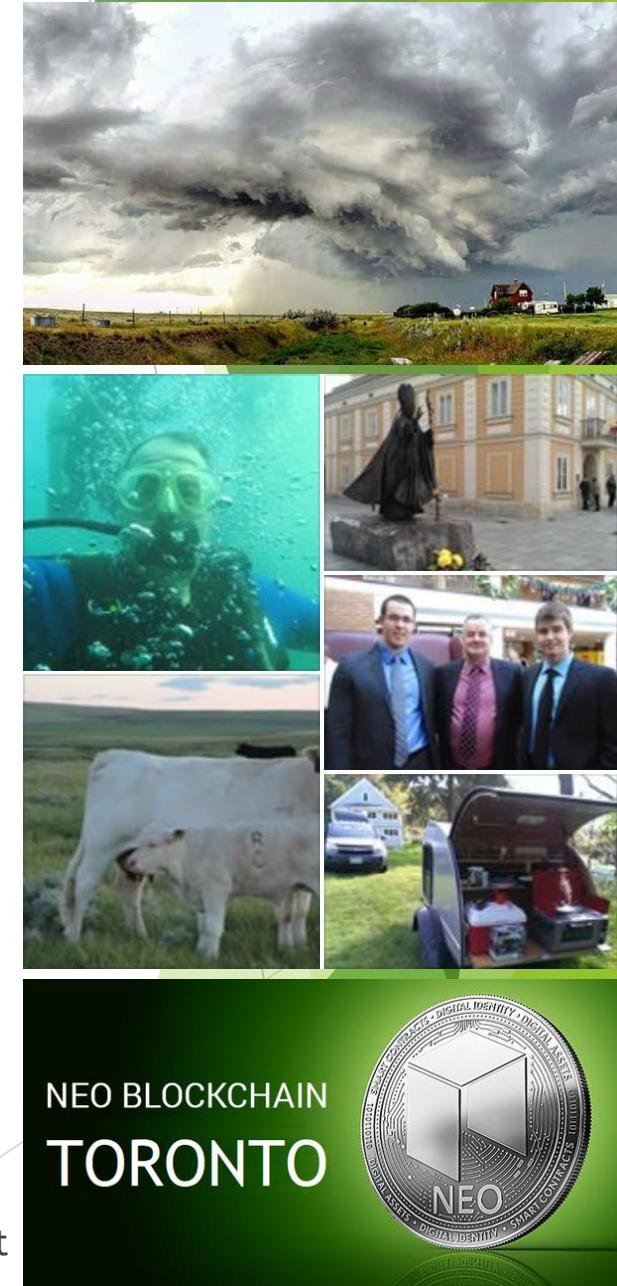
Detaylı Bilgi ve Başvuru >

blokzincir.bilgem.tubitak.gov.tr

2-3 Nisan 2018 ANKARA JW Marriott Hotel

Who is Michael Herman?

- ▶ Independent Blockchain Developer
 - ▶ Work 14 hours a day on the NEO Blockchain
 - ▶ Experienced Ethereum Developer
 - ▶ Work for myself and on behalf of my clients
- ▶ Decades of experience of software development experience
 - ▶ Every version of Microsoft Windows since the very first
- ▶ Curator of NEO C# Developers Center of Excellence community ([neo-csharp](#))
 - ▶ Contributes to several NEO community projects
 - ▶ NEO developer tool suite: [neo-debugger](#) and [neo-gui-developer](#) projects
 - ▶ NEO Persistable Class Framework (NPC) for efficient object-oriented smart contract development using C#.NEO: [mwherman2000/neo-persistentclasses](#) and [neo-npcc2s](#)
 - ▶ NEO Blockchain Quick Start Guide for .NET Developers: [mwherman2000/dotnetquickstart](#)
 - ▶ Founder of the first Canadian NEO Blockchain Meetup: NEO Blockchain Toronto



Who are you?

- ▶ Software Developer/Achitect?
 - ▶ Ethereum smart contract developers?
 - ▶ NEO smart contract developers?
 - ▶ Microsoft .NET developers?
 - ▶ Other blockchain developers?
- ▶ Crytopinvestor/trader?
- ▶ Legal profession?
- ▶ Financial services (conventional)?

- ▶ Government?
- ▶ Banking?
- ▶ Other sectors?

- ▶ Students?

What is the NEO Smart Economy?



=

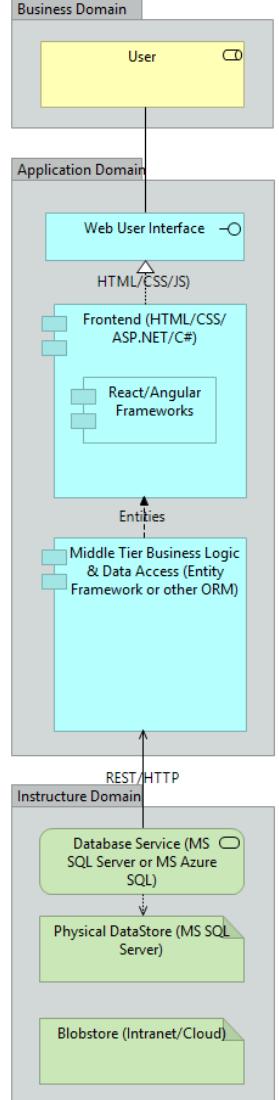


Digital Identity

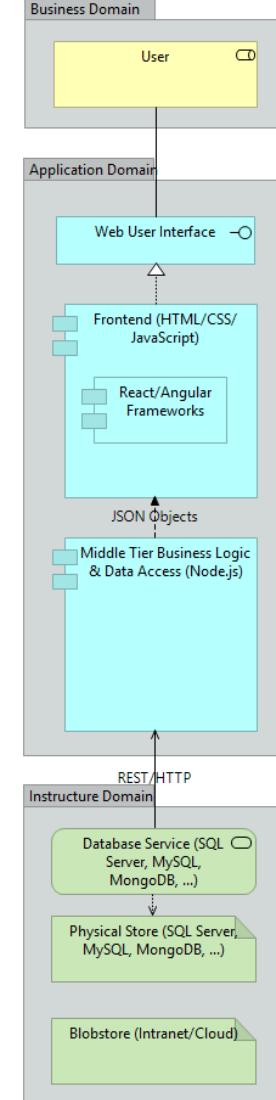
Future world where the boundary between real assets in the physical world and digital assets in the digital world has been removed.

Traditional Multi-Enterprise Application Examples

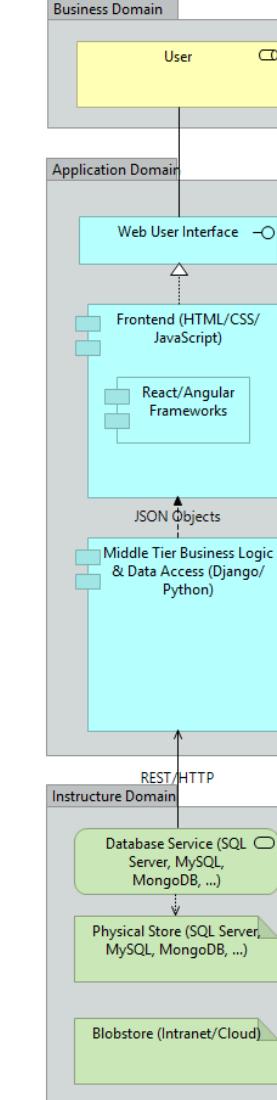
Microsoft ASP.NET/EF/SQL



Full Stack HTML/CSS/JS

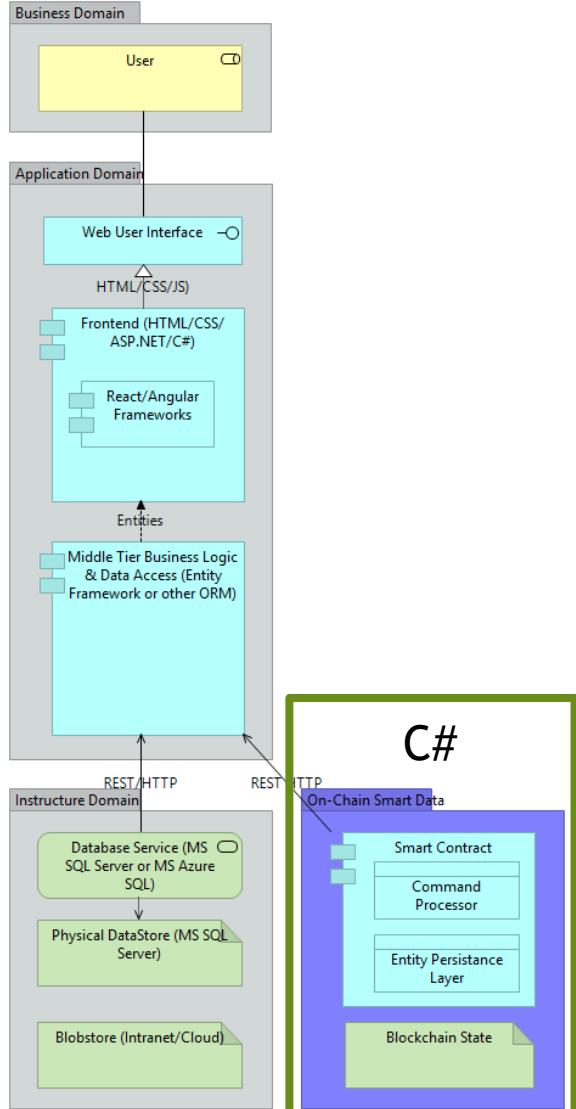


Full Stack HTML/CSS/Python

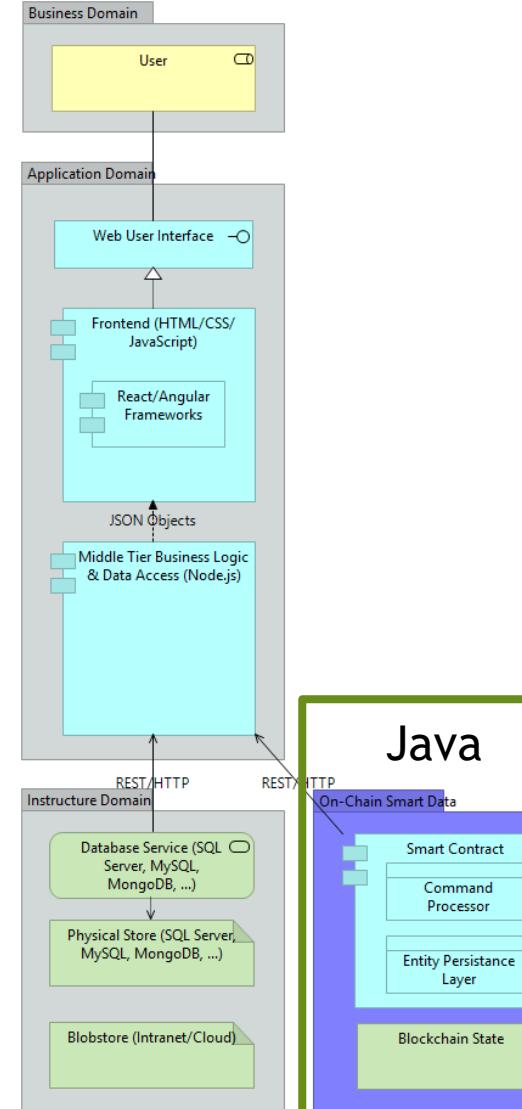


Enterprise Distributed Application Architecture

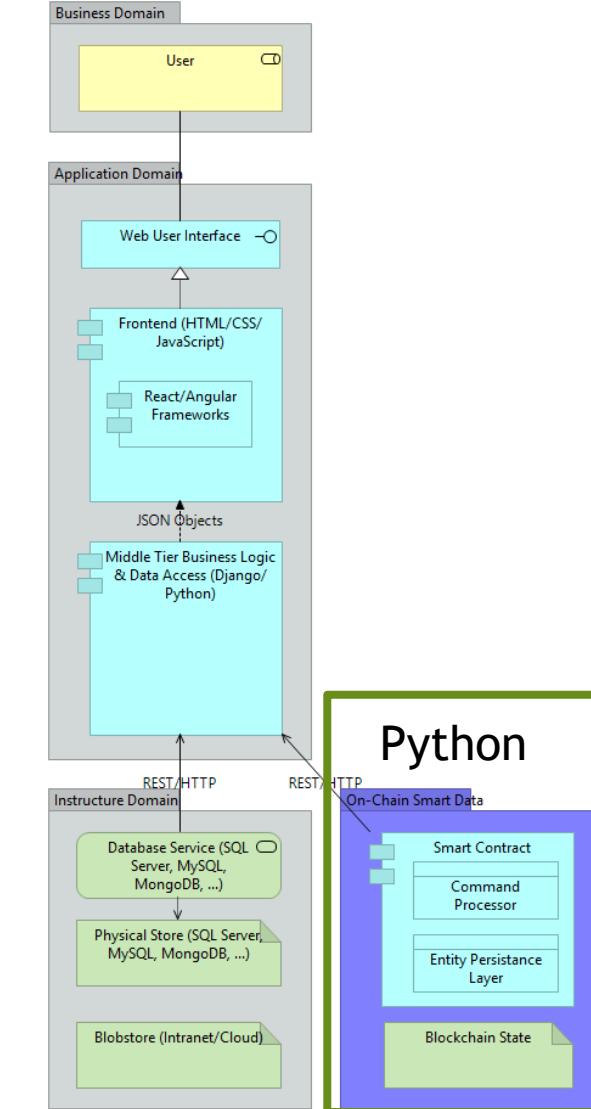
Microsoft ASP.NET/EF/SQL



Full Stack HTML/CSS/JS



Full Stack HTML/CSS/Python



When to use Blockchain Technology

“Blockchain: the force multiplier for the smart economy” [Microsoft whitepaper]

Questions to Ask

1. Is it a business process that crosses trust boundaries?
2. Do multiple parties manipulate the same data?
3. Are processes operating inefficiently or decisions delayed due to the number of intermediaries?
4. Does the business process involve low-value, manual verification steps?

Use a Blockchain when there is a Need for

- ▶ Real-time transparency by connecting business processes across organizations
- ▶ Real-time, transparent access to a verifiable source of the truth across organization boundaries
- ▶ Introduce trust and increase efficiency amongst participants - reducing the need for intermediaries
- ▶ Improved efficiencies and increased confidence through automation and smart contracts that execute consistently.

“Use blockchain technology when it matches the requirements of your application. Don’t automatically use blockchain technology simply because you’re re-platforming an existing application.” [Gartner 2018]

Purpose and Goals

- ▶ Purpose
 - ▶ Hands-on smart contract development workshop: .NET/C#/Visual Studio on the NEO Blockchain
- ▶ Goals, Non-goals and Assumptions
 - ▶ Target audience: Architects and Developers who are new to the NEO Blockchain platform
 - ▶ Background: Very little or no previous NEO or .NET/C#/Visual Studio Experience
 - ▶ Background: Understand basic 1st and 2nd generation blockchain technologies: hashing, blocks, mining, etc.
 - ▶ **Provide you with a basic level of awareness and understanding of NEO smart contract development**
 - ▶ **Help create a strong NEO Blockchain community in Turkey - helping each other**
- ▶ Principles
 - ▶ Provide reliable documentation: timely, accurate, visual, and complete
 - ▶ Save as much of a person's time as possible
 - ▶ Use open source software whenever possible
- ▶ Drivers
 - ▶ Need in the NEO .NET developer community to have concise and easy-to-follow documentation to enable people to get up to speed developing NEO smart contracts in as short a time as possible

Agenda

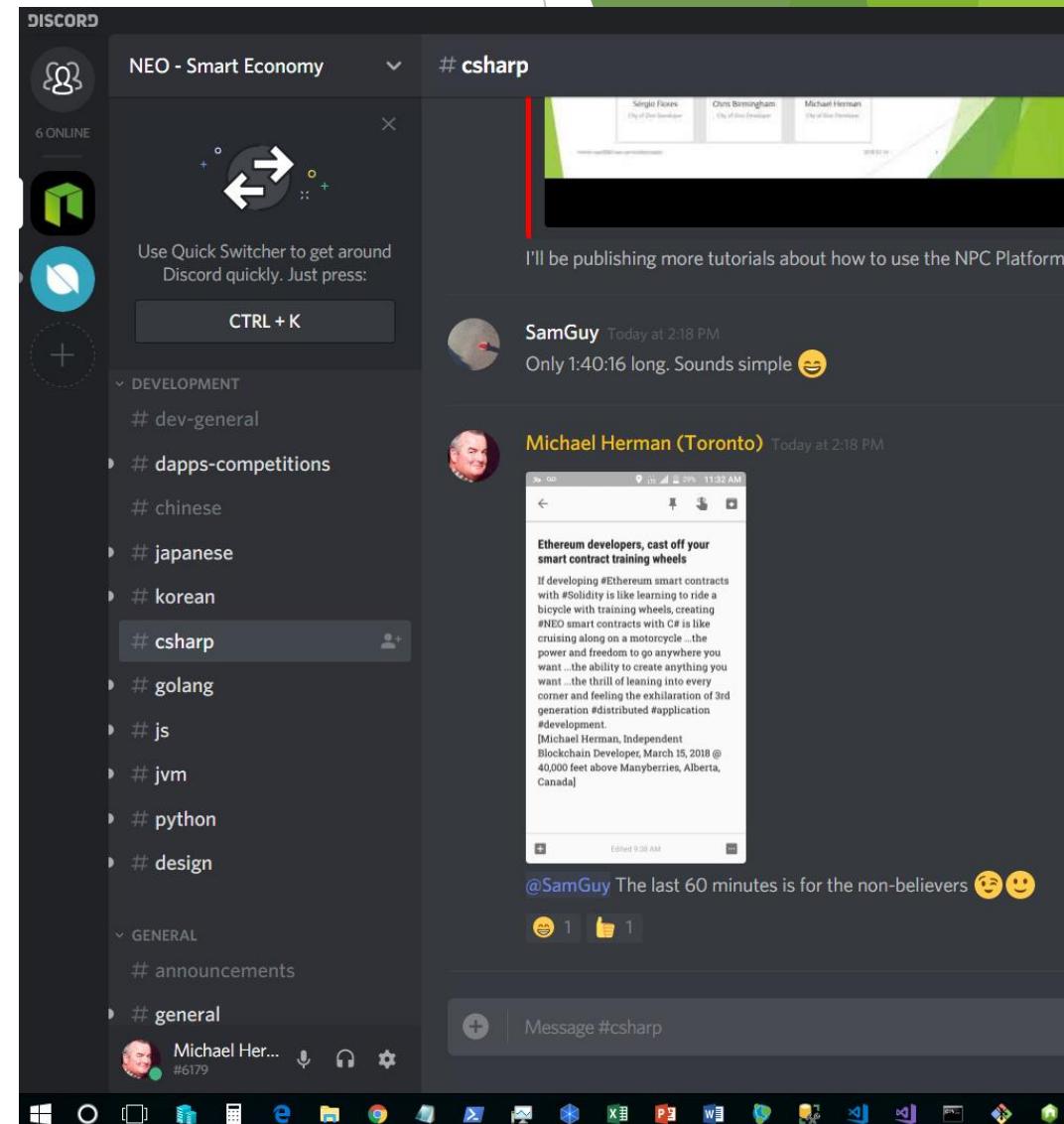
1. Introduction to NEO
2. NEO Whitepaper: Highlights for Developers
3. Why NEO?
4. Hands-on Lab: Creating a NEO Smart Contract in C#
5. NEO as a 3rd generation Enterprise Distributed Application Platform
6. NEO Evidence
7. NEO Resources
8. Next Steps

Module 1 Labs

Runtime.Log(), Runtime.Notify(), and NEO Storage

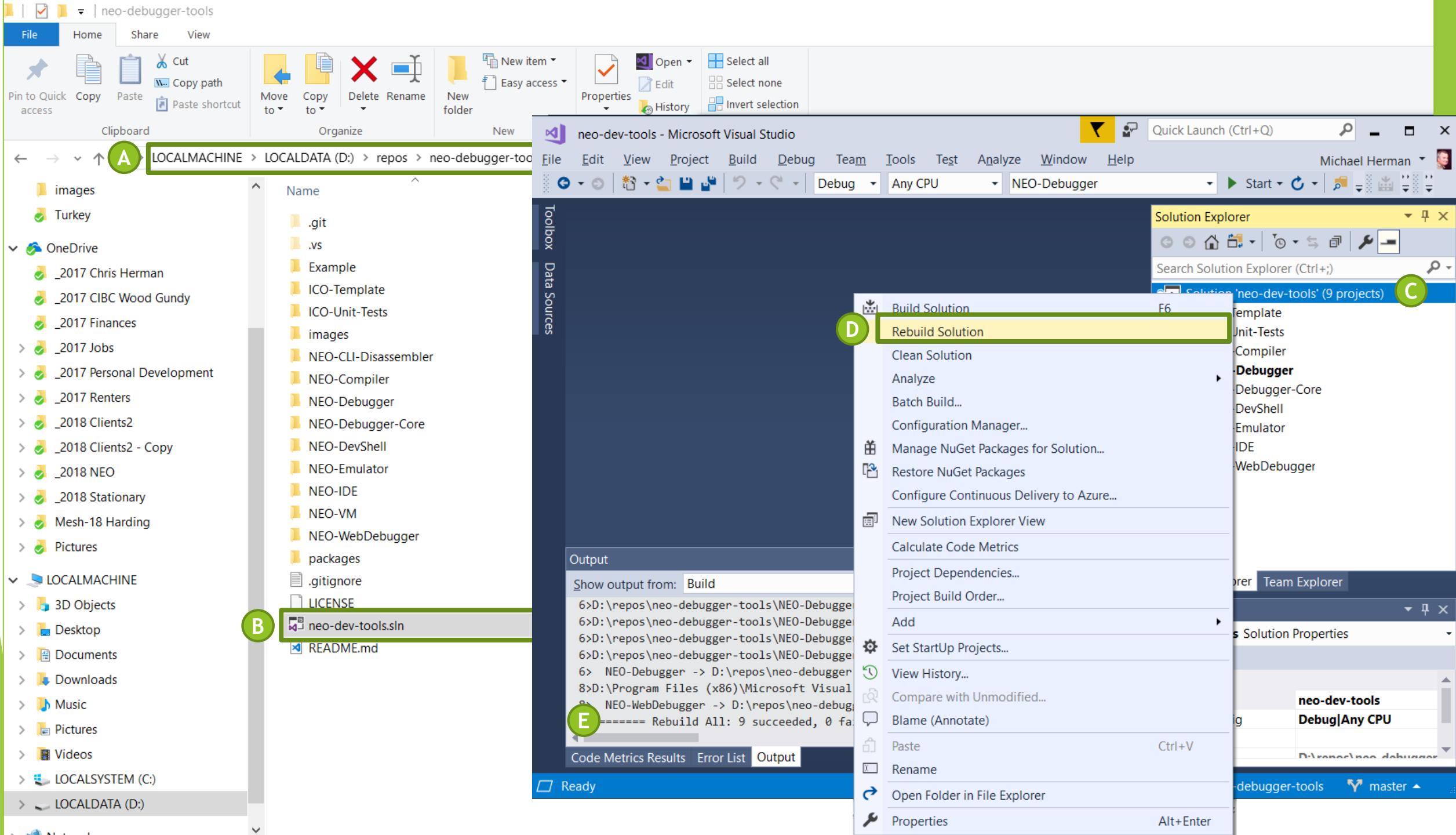
Preparation: Twitter, Discord and Visual Studio

1. I will be sending links to additional downloads and information using **#BlokchainTRDay** on Twitter
2. Join Discord NEO Developer Community on Discord by clicking on one of the following links:
<https://discord.gg/gqCYeup> or
<https://discord.gg/4TQujHj>
3. Download and install Microsoft Visual Studio Community Edition 2017. This version is free to download and free to use. It is an excellent IDE for C# smart contract development.
<https://www.visualstudio.com/vs/community/>



Preparation: NEO Debugger Tools

- ▶ Download the NEO Debugger Tools source code.
<https://github.com/mwherman2000/neo-debugger-tools/archive/master.zip>
 - ▶ Unzip and build the NEO Debugger Tools source code.
 - ▶ Follow the steps to rebuild the solution on the next slide.
-
- ▶ ****NOTE**** Please download a new copy of the master.zip file. It was updated last night.



Lab 1 - lab1_log.cs

“Hello World” Hands-on Lab



File Edit View Project Debug Team Tools Test Analyze Window Help

New ▾

- Project... Ctrl+Shift+N
- Repository...
- File... Ctrl+N
- Project From Existing Code...
- Spell Checker Configuration File for Selected Item

Open

Start Page

Close

Close Solution

Save Selected Items Ctrl+S

Save Selected Items As...

Save All Ctrl+Shift+S

Page Setup...

Print... Ctrl+P

Account Settings...

Recent Files ▾

Recent Projects and Solutions ▾

Exit Alt+F4

New Project



Recent

Installed

Visual C#

Windows Universal
Windows Classic Desktop

Web

.NET Core
.NET Standard

Android

Cloud

Cross-Platform

iOS

Test

tvOS

WCF

Visual Basic

Visual C++

Visual F#

SQL Server

Azure Data Lake

Machine Learning

Not finding what you are looking for?

[Open Visual Studio Installer](#)

Sort by: Default



Class Library (.NET Framework)

Visual C#



ASP.NET Core Web Application

Visual C#



ASP.NET Web Application (.NET Framework)

Visual C#



Shared Project

Visual C#



Class Library (Legacy Portable)

Visual C#



WCF Service Application

Visual C#



Azure Functions

Visual C#



Azure WebJob (.NET Framework)

Visual C#



Azure Cloud Service

Visual C#



Blank App (Android)

Visual C#



NeoContract

Visual C#



Wear App (Android)

Visual C#

Search (Ctrl+E)



Type: Visual C#

Allow users to create a CSharp project for NeoContract.

Name:

lab1

Location:

c:\users\mwhe...repos

[Browse...](#)

Solution name:

lab1

 Create directory for solution Create new Git repository

OK

Cancel



Contract1.cs

```
C# lab1
lab1.Contract1
Main()

1 using Neo.SmartContract.Framework;
2 using Neo.SmartContract.Framework.Services.Neo;
3 using System;
4 using System.Numerics;

5
6 namespace lab1
7 {
8     public class Contract1 : SmartContract
9     {
10         public static void Main()
11         {
12             Storage.Put(Storage.CurrentCont...
```

- Build Solution F6
- Rebuild Solution
- Clean Solution
- Analyze
- Batch Build...
- Configuration Manager...
- Manage NuGet Packages for Solution...
- Restore NuGet Packages
- New Solution Explorer View
- Calculate Code Metrics
- Add
- Set StartUp Projects...
- View History...
- Compare with Unmodified...
- Blame (Annotate)
- Paste Ctrl+V
- Rename
- Open Folder in File Explorer
- Properties Alt+Enter

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

Solution 'lab1' (1 project)

lab1

- Properties
- References
- build.tasks
- Contract1.cs
- Neo.ConvertTask.dll
- packages.config

Explorer Team Explorer

Solution Properties

config	Debug Any CPU
option	



lab1 - Microsoft Visual Studio



Quick Launch (Ctrl+Q)



File Edit View Project Build Debug Team Tools Test Analyze Window Help



Contract1.cs

```
1 using Neo.SmartContract.Framework;
2 using Neo.SmartContract.Framework.Services.Neo;
3 using System;
4 using System.Numerics;
5
6 namespace lab1
7 {
8     public class Contract1 : SmartContract
9     {
10         public static void Main()
11         {
12             Storage.Put(Storage.CurrentContext, "Hello", "World");
13         }
14     }
15 }
16
```

100 %

Code Metrics Results Error List

This item does not support previewing

Debug

Any CPU

Start

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

Solution 'lab1' (1 project)

lab1

- Properties
- References
- build.tasks
- Contract1.cs
- Neo.ConvertTask.dll
- packages.config

Solution Explorer

Team Explorer

Properties

lab1 Solution Properties



Misc

(Name)	lab1
Active config	Debug Any CPU
Description	

↑ 2

0

lab1

master ▾



```
1  -> using Neo.SmartContract.Framework;
2      using Neo.SmartContract.Framework.Services.Neo;
3      using System;
4      using System.Numerics;
5
6  -> namespace lab1_log
7  {
8      ->     public class lab1_log : SmartContract
9      {
10         ->         public static void Main()
11         {
12             Runtime.Log("Hello world!");
13         }
14     }
15 }
```

Lab 2 - lab2_logandnotify.cs

Runtime.Log() and Runtime.Notify()

```
1  using Neo.SmartContract.Framework;
2  using Neo.SmartContract.Framework.Services.Neo;
3  using System;
4  using System.Numerics;
5
6  namespace lab2_logandnotify
7  {
8      public class lab2_logandnotify : SmartContract
9      {
10         public static void Main()
11         {
12             // `Log()` is best to add simple strings to the NEO event log for development purposes
13             Runtime.Log("log 1");
14             Runtime.Log("log 2");
15             Runtime.Notify("notify 3");
16
17             // `Notify()` is best for adding variables such as lists and objects to the NEO event log
18             int ten = 10;
19             Runtime.Notify("ten", ten);
20
21             object[] results = { "a", 1, 2, "3" };
22             Runtime.Notify("results", results);
23
24             User e1 = new User { FirstName = "Able", LastName = "Baker" };
25             Runtime.Notify("e1", e1);
26             User e2 = new User { FirstName = "Charlie", LastName = "Delta" };
27             User[] userEntities = { e1, e2 };
28             Runtime.Notify("userEntities", userEntities);
29         }

```

```
30  
31     public class User  
32     {  
33         public string FirstName;  
34         public string LastName;  
35     }  
36 }
```

Lab 3 - lab3_storage.cs

NEO Storage

NEO Storage

- ▶ Each smart contact deployed to the NEO Blockchain has its own key-value data store
- ▶ Keys = byte[] array
- ▶ Values = string, byte[] array or BigInteger

Storage	X
Key	Content
84-99-21-A9-19-A3-1F-42-54-3A-8D-C3-64-3F-CB-9E-02-5F-20-FF-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-5F-53-54-41	3
84-99-21-A9-19-A3-1F-42-54-3A-8D-C3-64-3F-CB-9E-02-5F-20-FF-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-4C-61-73-...	0
84-99-21-A9-19-A3-1F-42-54-3A-8D-C3-64-3F-CB-9E-02-5F-20-FF-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-44-65-62-...	-100
84-99-21-A9-19-A3-1F-42-54-3A-8D-C3-64-3F-CB-9E-02-5F-20-FF-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-42-61-6C-...	999900
01-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-5F-53-54-41	3
01-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-4C-61-73-74-54-78-54-69-6D-65-73-74-61-6D-70	0
01-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-44-65-62-69-74-43-72-65-64-69-74-41-6D-6F-75-6E-74	100
01-2F-23-4E-50-43-4E-45-50-35-4C-65-64-67-65-72-45-6E-74-72-79-2E-42-61-6C-61-6E-63-65	100

NEO Storage API

Storage [from metadata] ■ ✚ ✖

C# Neo.SmartContract.Framework Neo.SmartContract.Framework.Services.Neo.Storage

```
7  namespace Neo.SmartContract.Framework.Services.Neo
8  {
9      public static class Storage
10     {
11         public static StorageContext CurrentContext { get; }
12
13         ...
14         public static void Delete(StorageContext context, byte[] key);
15         ...
16         public static void Delete(StorageContext context, string key);
17         ...
18         public static byte[] Get(StorageContext context, byte[] key);
19         ...
20         public static byte[] Get(StorageContext context, string key);
21         ...
22         public static void Put(StorageContext context, byte[] key, byte[] value);
23         ...
24         public static void Put(StorageContext context, byte[] key, BigInteger value);
25         ...
26         public static void Put(StorageContext context, byte[] key, string value);
27         ...
28         public static void Put(StorageContext context, string key, byte[] value);
29         ...
30         public static void Put(StorageContext context, string key, BigInteger value);
31         ...
32         public static void Put(StorageContext context, string key, string value);
33     }
34 }
```

```
1  using Neo.SmartContract.Framework;
2  using Neo.SmartContract.Framework.Services.Neo;
3  using System;
4  using System.Numerics;
5
6  namespace lab3_storage
7  {
8      public class lab3_storage : SmartContract
9      {
10         public static void Main()
11         {
12             // Get a copy of the NEO Storage context
13             StorageContext ctx = Storage.CurrentContext;
14
15             string item_key = "test-storage-key";
16             Runtime.Notify("item_key", item_key);
17
18             // Try to get a value for this key from storage
19             BigInteger item_value = Storage.Get(ctx, item_key).AsBigInteger();
20             Runtime.Notify("item_value", item_value);
21             if (item_value == 0)
22             {
23                 Runtime.Notify("Storage key not set. Setting item_value to 1");
24                 item_value = 1;
25             }
26             else
27             {
28                 Runtime.Notify("Storage key already set. Incrementing item_value by 1");
29                 item_value += 1;
30             }
31
32             // Put the updated value for this key into NEO Storage
33             Storage.Put(ctx, item_key, item_value);
34             Runtime.Notify("New item_value", item_value);
35         }
36     }
37 }
```

```
10     public static void Main()
11     {
12         // Get a copy of the NEO Storage context
13         StorageContext ctx = Storage.CurrentContext;
14
15         string item_key = "test-storage-key";
16         Runtime.Notify("item_key", item_key);
17
18         // Try to get a value for this key from storage
19         BigInteger item_value = Storage.Get(ctx, item_key).AsBigInteger();
20         Runtime.Notify("item_value", item_value);
21         if (item_value == 0)
22         {
23             Runtime.Notify("Storage key not set. Setting item_value to 1");
24             item_value = 1;
25         }
26         else
27         {
28             Runtime.Notify("Storage key already set. Incrementing item_value by 1");
29             item_value += 1;
30         }
31
32         // Put the updated value for this key into NEO Storage
33         Storage.Put(ctx, item_key, item_value);
34         Runtime.Notify("New item_value", item_value);
35     }
36 }
```

End of Module 1

- ▶ Lab 1 - lab1_log.cs
- ▶ Lab 2 - lab2_logandnotify.cs
- ▶ Lab 3 - lab3_storage.cs

1. What is NEO?

What is NEO?

- ▶ PROJECT: NEO (<https://neo.org/>) is a non-profit, community-based blockchain project that utilizes blockchain technology and digital identity to digitize assets, to automate the management of digital assets using smart contracts, and to realize a "smart economy" with a distributed network.
- ▶ SOFTWARE PLATFORM: 3rd generation distributed applications (dApp) platform
 - ▶ Multiple languages: C# (reference implementation), Python, Java, Golang, JavaScript
 - ▶ First dApp platform suitable for enterprise application development, deployment, and operations
- ▶ DUAL CRYPTOCURRENCY: NEO and GAS
- ▶ NEO SMART ECONOMY
 - ▶ Key differentiator
- ▶ GLOBAL COMMUNITY
 - ▶ Discord: <https://discord.gg/gqCYeup> / <https://discord.gg/4TQujHj>
 - ▶ NEO C# Developers Center of Excellence (neo-csharpcoe)
<https://github.com/mwherman2000/neo-csharpcoe/blob/master/README.md>
 - ▶ NEL (NewEcoLab) “Chinese-speaking NEO Community”
<https://github.com/neweconolab>
 - ▶ City of Zion “Home of the Open-source Global NEO Developer Community”
<https://cityofzion.io/>

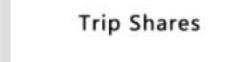
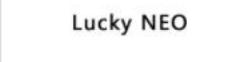
Dual Cryptocurrency Model: NEO and NeoGas

- ▶ NEO Smart Economy, there is a separation of concerns:
 - ▶ NEO represents Voting Power
 - ▶ NeoGas represents Ability to Fund Work (Systems Fees)
- ▶ NEO (NEO)
 - ▶ Intended use: acquire, hold, and retain NEO to maximize your voting power in the network
 - ▶ Receive distributions of GAS
- ▶ NeoGas (GAS)
 - ▶ Sometimes referred to as a “Utility Token” or the “Fuel Token”
 - ▶ Used to fund the deployment of smart contracts (applications) on the NEO Blockchain
 - ▶ Used to fund transactions executed on the NEO Blockchain
 - ▶ System Fees are recycled and re-distributed in ratio/proportion to the number of NEO you hold

NEO dApp List

These dAPPS come from the community. The information on this page is provided by the project owner. NEO Council and ndapp.org does not endorse these projects.

Add a NEO dApp

 NEX by The NEX Team	 Moonlight by The Moonlight Team	 Red Pulse by Red Pulse	 AdEx by adexnetwork	 NeoAuth by @NeoAuth	 ZEEPIN by ZEEPIN	 Qlink by Alkin Li	 Neo Smart IoT by haito2929, jpharmer	 imusify by DavidWalters125, geeklife, Nikolaj K, metachris
NEX combines the NEO blockchain with an off-chain matching engine to enable much faster and more complex trades than existing decentralized exchanges	Moonlight is a distributed workforce and analytical project management platform featuring a global public ledger of contributor work experience and a new match-making algorithm to effectively fulfill project needs.	Red Pulse Tokens (RPX) are NEO tokens issued by Red Pulse, an event-driven research firm covering market events impacting Chinese companies, sectors and the overall economy.	AdEx is a decentralized ad exchange built on the blockchain and smart contracts. The core feature of AdEx will be the so-called AdEx User Profile - a personalized page that allows every end user to understand and control the ads delivered to them.	NeoAuth enables authentication over the NEO blockchain, allowing you to log in with a NEO address instead of an email and password.	The Distributed Creative New Economy. Zeepin, a decentralized innovation community, is dedicated to promoting highly efficient circulation of innovation assets. Smart work. Creative life!	Qlink, developed by Qlink Foundation in Singapore, adopts the blockchain technology and creates a decentralized mobile network for P2Peer WiFi sharing, mobile data converted content distribution, enterprise telecom services and crowd-sourcing base stations (including	Control IoT (Internet of Things) devices via Neo smart contracts (first device is an ESP8266).	imusify is a free, blockchain based, incentivized and decentralized platform for music related digital content such as audio, video, apps, images, and blogging where anyone can join, contribute and get paid \$IMU.
Website Whitepaper Twitter LinkedIn Medium	Website	Website GitHub Reddit Contract	Website GitHub Medium Facebook Telegram Twitter Reddit	Website Demo GitHub Whitepaper	Website Telegram Medium GitHub Contract	GitHub Medium Page AMA summary Contract	Website GitHub Contract	Website GitHub
 Chain Line by notatotuser	 Phantasma by Raffiti	 NeoTrade by birmas	 Turing Complete Smart Contract by Nikolaj K	 KRYPTON.live by birmas	 Switcheo by RavenKos, hennychea, jackyliu	 Trip Shares by etolya	 BlockSaver by imprunetc, morigan	 Lucky NEO by mmoravec, horang, StefanBach, WesleyLeung, tamersocial
Peer-to-peer courier platform. Couriers transport items to fulfill demands and earn courier fees.	Phantasma is a platform where the users control their own content, instead of relying in third parties servers. The platform support any kind of transactions between users, e.g. email, chat, files, money transfers. NEO was used for this project due to its fast transactions and C#.	NeoTrade is a NEP5 trading platform inspired by Etherdelta. NeoTrade is a NEO based smart contract that allows users to deposit NEO, GAS or any other NEP-5 asset directly to the smart contract and trade amongst themselves.	A smart contract which interprets encoded classical Turing machines.	KRYPTON is a Smart Contract and ecosystem to provide SIP registration on the NEO blockchain using a KRYPTON compatible SIP provider.	Switcheo is a decentralised exchange built on NEO's blockchain. It supports trading of NEP-5 tokens and NEO system assets like GAS & NEO. It aims to be a truly decentralised exchange while still providing for a superb user experience.	Trip sharing with deposits. Travellers commit to a shared seat by depositing \$TRS NEP-5 tokens, which are used as insurance when the passenger cancels the trip after a set date.	Saving smart contract, with interests and fines. You can define how long to lock NEO and GAS assets, the smart contract defines an interest rate, and the system has a penalty for early withdrawal.	Lucky NEO allows anyone to send their extra gas to a raffle contest. One winner will be chosen every two weeks and automatically paid out. Lucky NEO uses an admin account to automatically pay out the winner to the address that sent the funds; so you don't have to follow
Website GitHub	Demo Website GitHub	Website GitHub	Youtube GitHub	Youtube Demo GitHub	Youtube Website GitHub	Website GitHub	GitHub	Post GitHub
 Neo Fund	 Neo Raffle by ambethia	 Nep Swap by block_shaman	 Smart Promise by SergeyAnkrumko, marlyonya	 Sunny dApp by jannvanderberg	 INS by NewEconLab(TEL)	 THEKEY by THEKEY team	 蓝鲸淘 by 中科技	 TRINITY by Trinity team
Neo Fund is a decentralized crowdfunding platform, similar to Kickstarter. The basic function is to set a goal amount, and date limit. If the goal is reached, the contributors will be awarded the funds; and if not the contributors can redeem their funds again.	A smart contract that enables a raffle/lottery on the Neo block chain. Send GAS, win GAS! This project also demonstrates a solid way to use random numbers in a smart contract.	Protocol and smart contract for trading NEP-5 based assets.NepSwap is a protocol and smart contract for trading NEP-5 assets. This type idea is based on EtherDelta and Bancor. It doesn't have a personal token or ICO, this project is intended for community, and	Smart Promise is an electronic journal of smart promises developed through Blockchain. User of such an environment receives a reward for each of his fulfilled promises. The concept of the idea contains the possibility to get a strong motivation to action that user pointed in	This dApp allows you to insure against bad weather conditions on a given day. If the relative sunshine duration on that day is lower than 50 percent, you get paid.	Distributed domain name service ending with .neo based on NEO blockchain.	THEKEY is a Decentralized Ecosystem of Identity Verification Tool Using National Big-data and Blockchain. THEKEY Project Team is now developing IDV identification verification (IDV) tool with blockchain based dynamic multi-dimension	蓝鲸淘是链资产中心化管理平台，通过开源的区块链技术与保险金的规则，为用户提供点对点、无中介的数字资产流转服务，降低数字资产流转过程中的信用成本。	Trinity is applicable to blockchain transfer of NEP-5 standard tokens. As a Neo-version lightning network, Trinity achieves real-time payment, low transaction fee scalability, and privacy protection of Neo mainnet assets through state channels
GitHub	Website GitHub	GitHub	Youtube Website GitHub Whitepaper	Website GitHub	Website GitHub	Website GitHub telegram facebook twitter Contract	Website GitHub	website GitHub telegram twitter Contract

NEO Has All Ingredients



High TPS: 1,000 TPS



Finality: 1 Confirmation, dBFT



Interoperability: NeoX



Digital ID: NeoID / Ontology



Stable Coin: Alchemint

By 2020, NEO Will Handle

100,000 TPS
Without Sharding!

NEO C# .NET Developers Center of Excellence



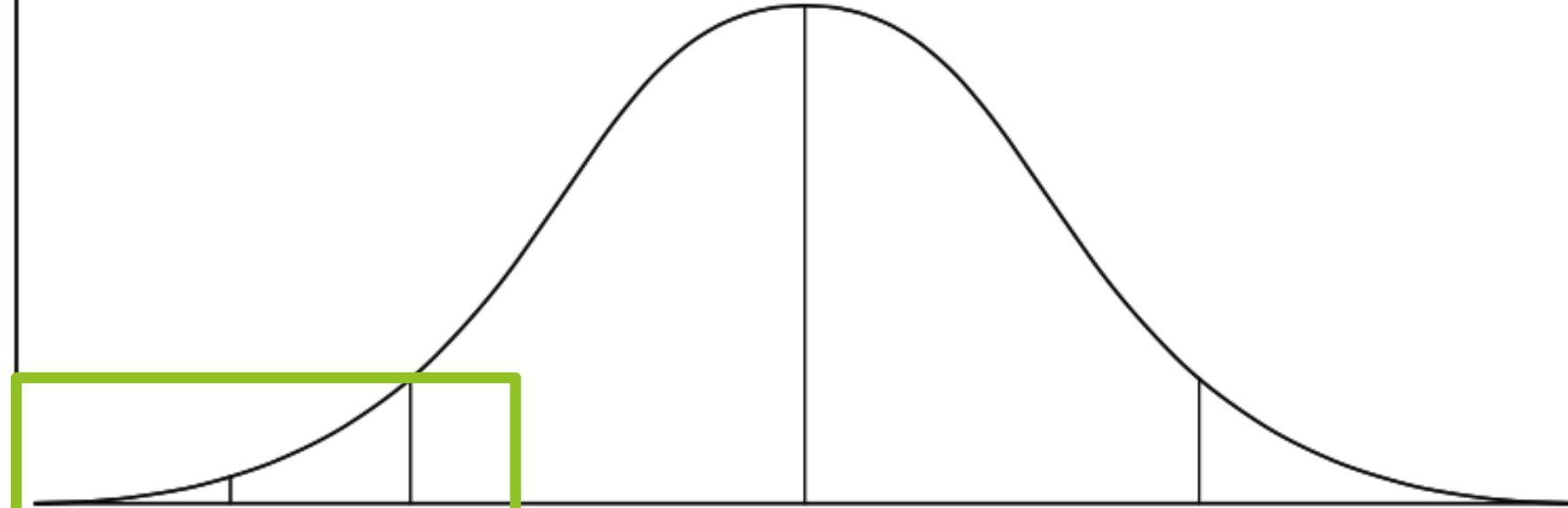
NEO Decentralized Applications (Use Cases)

- ▶ Smart fund
- ▶ AI-assisted legal smart contracts
- ▶ Social networking
- ▶ Automated token liquidity providers
- ▶ Decentralized exchanges
- ▶ Secure communications protocol
- ▶ Supply chain (parts, food, wine, ...)
- ▶ Consortiums (business partnerships)
- ▶ Data exchange markets
- ▶ Intellectual property trading markets
- ▶ Prediction markets
- ▶ Advertising markets
- ▶ Hashpower (cryptohealth) markets
- ▶ NeoGas markets
- ▶ Business process re-engineering
- ▶ Requisition processing

Blockchain: Early Adopters Marketplace

The Technology Adoption Curve

*As captured by Everett Rogers in his book *Diffusion of Innovations*, people tend to adopt new technologies at varying rates. Their relative speed of adoption can be plotted as a normal distribution, with the primary differentiator being individuals' psychological disposition to new ideas.*



Innovators

(2.5%) are risk takers who have the resources and desire to try new things, even if they fail

Early Adopters

(13.5%) are selective about which technologies they start using. They are considered the “one to check in with” for new information and reduce others’ uncertainty about a new technology by adopting it.

Early Majority

(34%) take their time before adopting a new idea. They are willing to embrace a new technology as long as they understand how it fits with their lives.

Late Majority

(34%) adopt in reaction to peer pressure, emerging norms, or economic necessity. Most of the uncertainty around an idea must be resolved before they adopt.

Laggards

(16%) are traditional and make decisions based on past experience. They are often economically unable to take risks on new ideas.

2. NEO Architecture

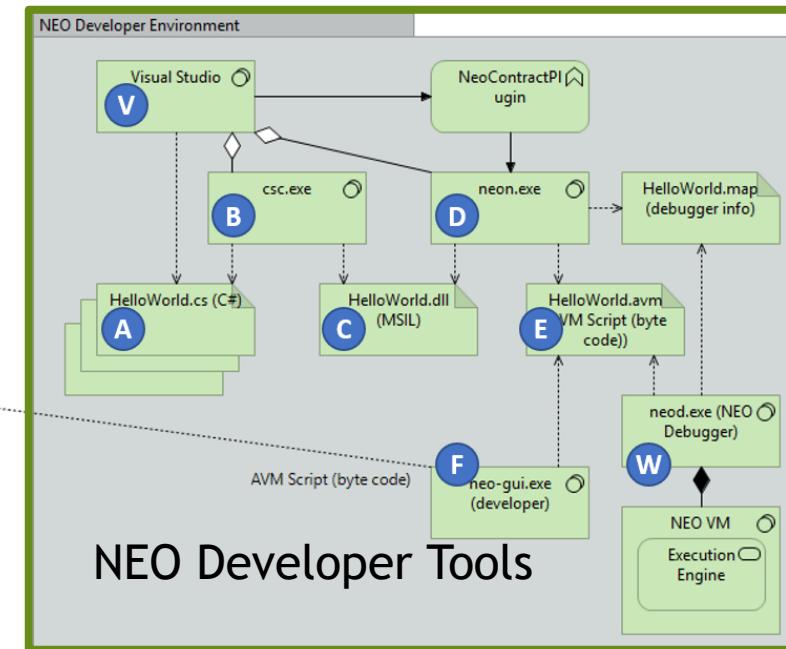
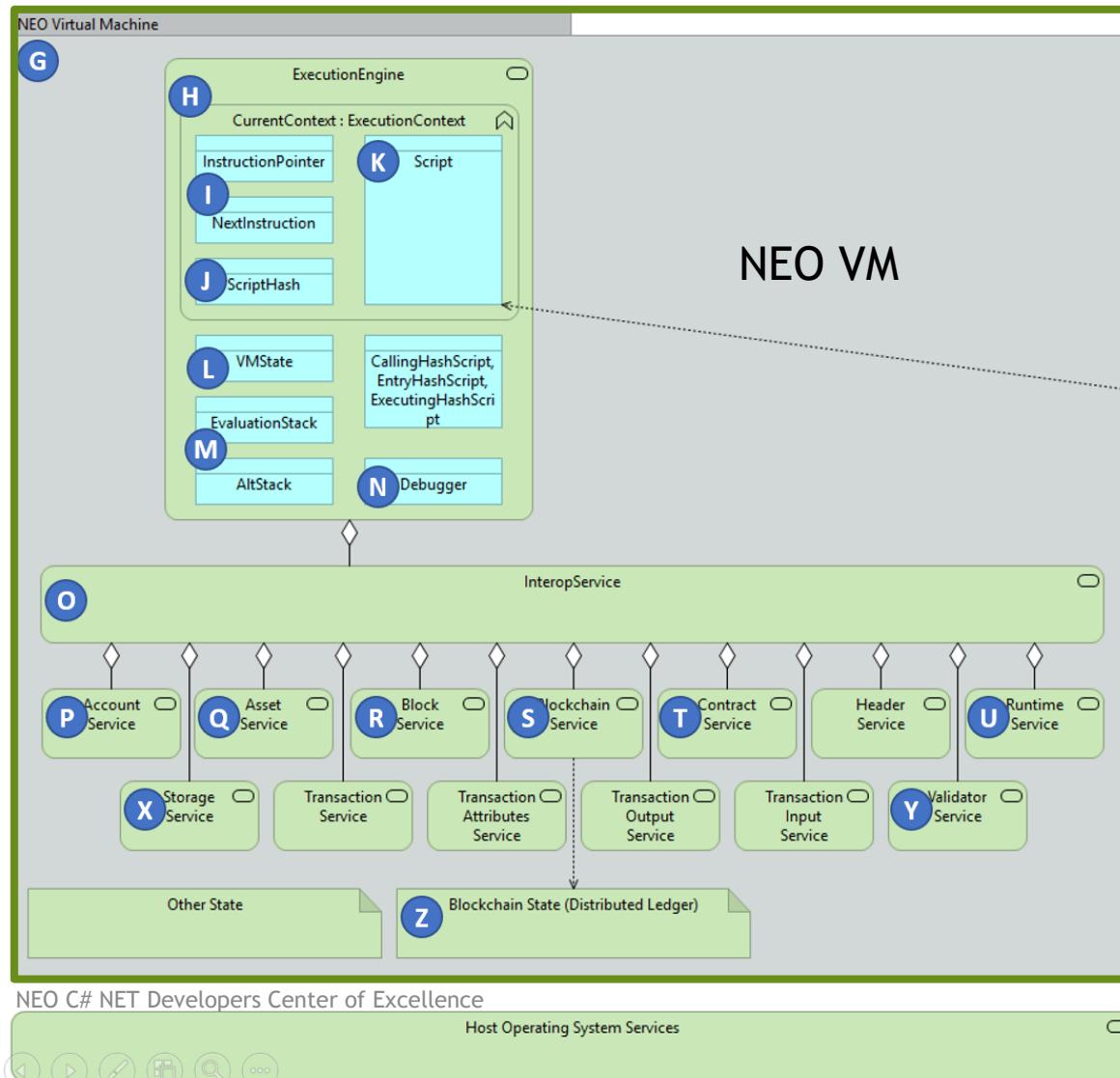
NEO Whitepaper

BREAK - 15 MINUTES

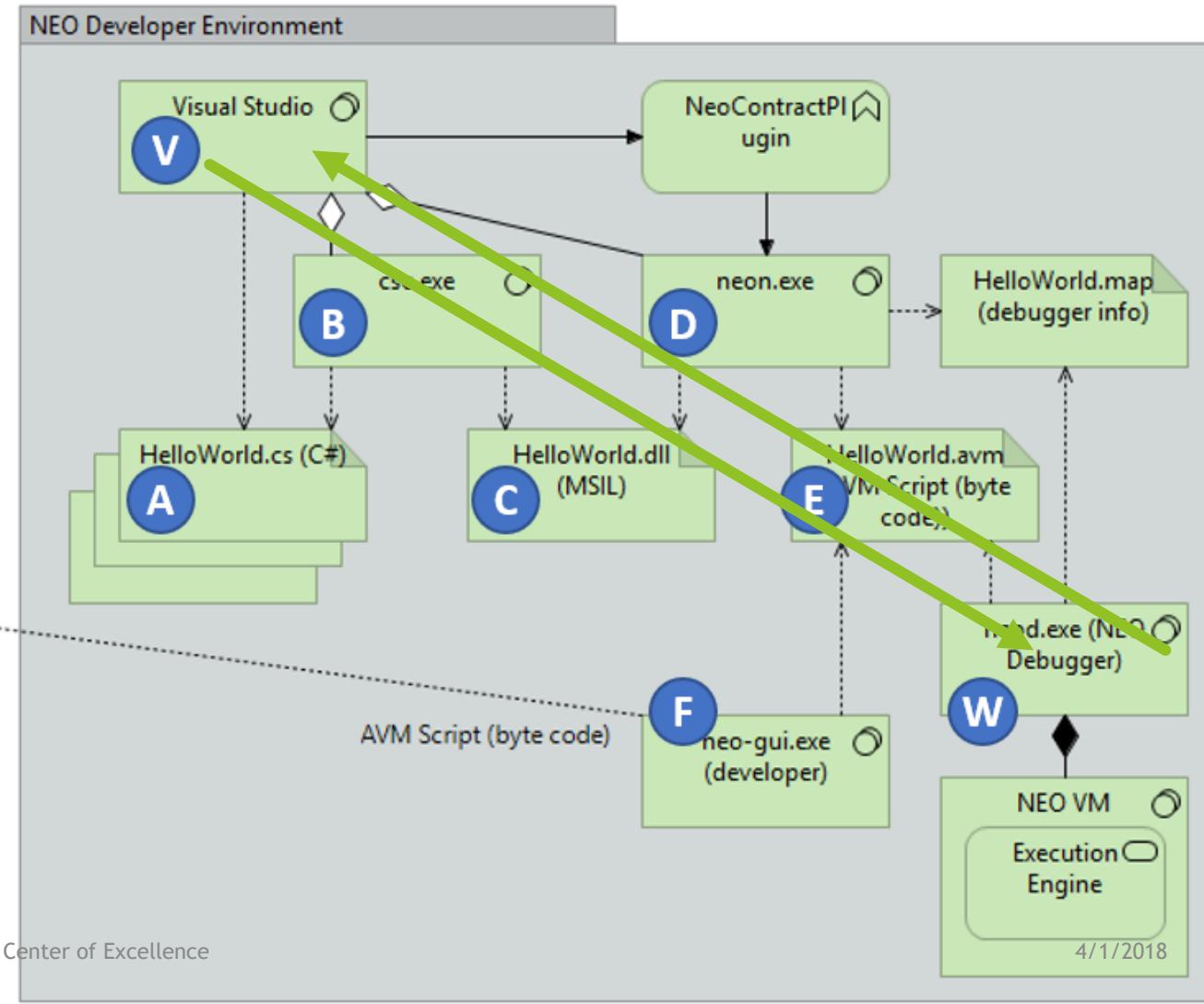


NEO Architecture

NEO Reference Architecture (neo-charm)



NEO Reference Architecture: Developer



C# Source Code
.cs

csc.exe

Microsoft Intermediate Language (.dll)

neon.exe

NEO VM Byte Code (.avm)

neod.exe

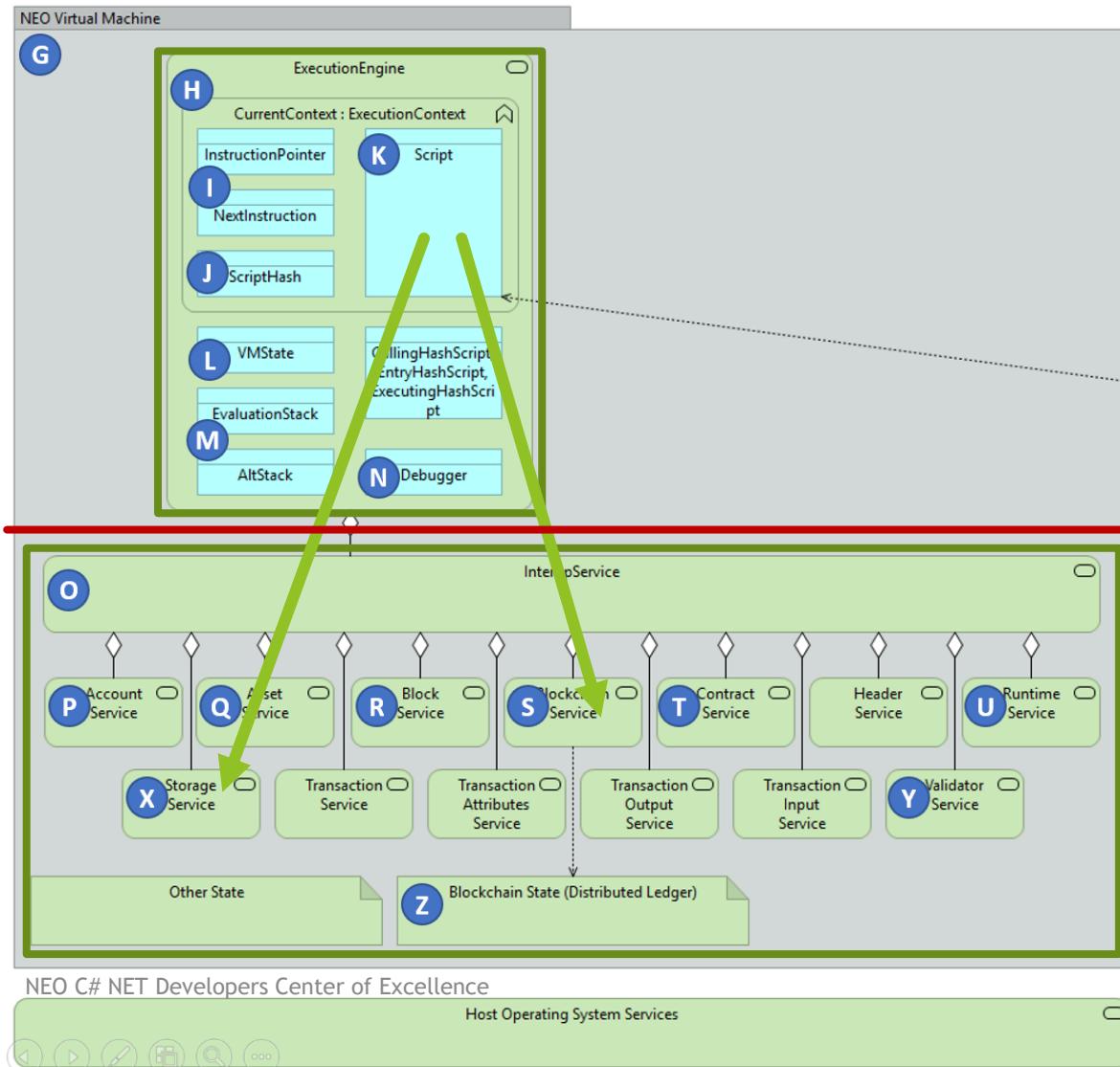
NEO VM

NEO System Fees

<http://docs.neo.org/en-us/sc/systemfees.html>

System Calls	NEO [Gas]	Instruction OpCodes	NEO [Gas]
Runtime.CheckWitness	0.2	OpCode.PUSH16 [or less]	0.00
Blockchain.GetHeader	0.1	OpCode.NOP	0.00
Blockchain.GetBlock	0.2	OpCode.APPCALL	0.01
Blockchain.GetTransaction	0.1	OpCode.TAILCALL	0.01
Blockchain.GetAccount	0.1	OpCode.SHA1	0.01
Blockchain.GetValidators	0.2	OpCode.SHA256	0.01
Blockchain.GetAsset	0.1	OpCode.HASH160	0.02
Blockchain.GetContract	0.1	OpCode.HASH256	0.02
Transaction.GetReferences	0.2	OpCode.CHECKSIG	0.10
Account.SetVotes	1	OpCode.CHECKMULTISIG [per signature]	0.10
Validator.Register	1000	Other OpCodes	0.001
Asset.Create (system asset)	5000	• Constants	
Asset.Renew (system asset)	5000	• Flow Control	
Contract.Create	500	• Stack	
Contract.Migrate	500	• String	
Storage.Get	0.1	• Logical	
Storage.Put [per KB]	1.0	• Arithmetic	
Storage.Delete	0.1	• Cryptography	
Other OpCodes	0.001	• Structure	
		• Exceptions	

NEO Reference Architecture: NEO Virtual Machine



System Calls	NEO [Gas]
Runtime.CheckWitness	0.2
Blockchain.GetHeader	0.1
Blockchain.GetBlock	0.2
Blockchain.GetTransaction	0.1
Blockchain.GetAccount	0.1
Blockchain.GetValidators	0.2
Blockchain.GetAsset	0.1
Blockchain.GetContract	0.1
Transaction.GetReferences	0.2
Account.SetVotes	1
Validator.Register	1000
Asset.Create (system asset,	5000
Asset.Renew (system asset)	5000
Contract.Create	500
Contract.Migrate	500
Storage.Get	0.1
Storage.Put [per KB]	1.0
Storage.Delete	0.1
Other OpCodes	0.001

Module 2 Labs

NEO Smart Contract Patterns: Process Operation, Domain Name Server, NEP5 Token

Lab 4 - lab4_processoperationpattern.cs

NEO Smart Contract Patterns

```
6  namespace lab4_processoperationpattern
7  {
8      public class lab4_processoperationpattern : SmartContract
9      {
10         public static object Main(string operation, params object[] args)
11         {
12             object result = false; // = 0 (zero)
13
14             if (args.Length == 0)
15             {
16                 Runtime.Log("Missing parameters");
17                 result = false;
18             }
19             else
20             {
21                 if (operation == "query")
22                 {
23                     Runtime.Notify("query", args[0]);
24                     result = args[0];
25                 }
26                 else if (operation == "delete")
27                 {
28                     Runtime.Notify("delete", args[0]);
29                     result = true;
30                 }
31                 else if (operation == "register")
32                 {
33                     Runtime.Notify("register", args[0]);
34                     result = true;
35                 }
36                 else if (operation == "transfer")
37                 {
38                     Runtime.Notify("transfer", args[0]);
39                     result = true;
40                 }
41                 else
42                 {
43                     result = false;
44                 }
45             }
46             return result;
47         }
48     }
```

Process Operation Pattern

```
public static object Main(string operation,  
                         params object[] args)
```

Parameters

- ▶ **operation** task to be performed by the smart contract
- ▶ **args** parameters need for a specific task

Return Value

- ▶ **object** bool, simple value, entity, array of objects

```
public static object Main(string operation, params object[] args)  
{  
    object result = false; // = 0 (zero)  
  
    if (args.Length == 0)  
    {  
        Runtime.Log("Missing parameters");  
        result = false;  
    }  
    else  
    {  
        if (operation == "query")  
        {  
            Runtime.Notify("query", args[0]);  
            result = args[0];  
        }  
        else if (operation == "delete")  
        {  
            Runtime.Notify("delete", args[0]);  
            result = true;  
        }  
        else if (operation == "register")  
        {  
            Runtime.Notify("register", args[0]);  
            result = true;  
        }  
        else if (operation == "transfer")  
        {  
            Runtime.Notify("transfer", args[0]);  
            result = true;  
        }  
        else  
        {  
            result = false;  
        }  
    }  
}  
return result;
```

```
public static object Main(string operation, params object[] args)
{
    object result = false; // = 0 (zero)

    if (args.Length == 0)
    {
        Runtime.Log("Missing parameters");
        result = false;
    }
    else
    {
        if (operation == "query")
        {
            Runtime.Notify("query", args[0]);
            result = args[0];
        }
        else if (operation == "delete")
        {
            Runtime.Notify("delete", args[0]);
            result = true;
        }
        else if (operation == "register")
        {
            Runtime.Notify("register", args[0]);
            result = true;
        }
        else if (operation == "transfer")
        {
            Runtime.Notify("transfer", args[0]);
            result = true;
        }
        else
        {
            result = false;
        }
    }
    return result;
}
```

Lab 5 - lab5_domain.cs

NEO Smart Contract Patterns

NEO Account Addresses, Keys and Key Lengths

Description	Length (byte[]/HexString)	Examples (Bold Lengths)
Address1	(34/68)	AcCHOikUq9cP6SMESHufCEMwADJNcTwnAv
Address1 WIF	(52/104)	L3f7C21q4Mu5FzZsDuCMeHqwJ1apHYCrwzU2821p1opaM43BAMKo
Address1 PublicKeyHex	(33/66)	02c44534465c8b21f659eba5708e69edae1ddd6f8cd63004095f8e39493cf54e82
Address1 PrivateKeyHex	(32/64)	c016e1c8a193cc1a28a15464106b91b52727547a3a36f40a8bfebd9933d1963c
Address1 ScriptHash	(20/40)	e000aa6a0ab08af8aa78b19d481e5b5c40d8be0e = Address1.AsScriptHash();
Address2	(34/68)	AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y
Address2 WIF	(52/104)	KxDgvEKzgSBPPfuVfw67oPQBSjidEiqTHURKSDL1R7yGaGYAeYnr
Address2 PublicKeyHex	(33/66)	031a6c6fbfdf02ca351745fa86b9ba5a9452d785ac4f7fc2b7548ca2a46c4fcf4a
Address2 PrivateKeyHex	(32/64)	1dd37fba80fec4e6a6f13fd708d8dcb3b29def768017052f6c930fa1c5d90bbb
Address2 ScriptHash	(20/40)	23ba2703c53263e8d6e522dc32203339dcd8eee9 = Address2.AsScriptHash();
TxID, AssetID	(32/64)	687b68a1159429dc558e4fc7590e391d52f1ef79a12922f941daa37c00334ec5

```
1  using Neo.SmartContract.Framework;
2  using Neo.SmartContract.Framework.Services.Neo;
3  using System;
4  using System.Numerics;
5
6  //Testing:
7  // operation,    args
8  // -----, -----
9  // "query",      ["test.com"]
10 // "register",   ["test.com", "AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y"]
11 // "delete",     ["test.com"]
12 // "transfer",   ["test.com", "AK2nJJpJr6o664CWJKi1QRXjqeic"]
13
14 namespace lab5_domain
15 {
16     public class lab5_domain : SmartContract
17     {
18         public static object Main(string operation, params object[] args)
19         {
```

lab5_domain.cs* X

C# lab5_domain lab5_domain

```
18     public static object Main(string operation, params object[] args)
19     {
20         object result = false; // = 0 (zero)
21
22         if (args.Length == 0)
23         {
24             Runtime.Log("No domain named supplied");
25             result = 0;
26         }
27         else
28         {
29             string domain_name = (string)args[0];
30             if (operation == "query")
31             {
32                 result = QueryDomain(domain_name);
33             }
34             else if (operation == "delete")
35             {
36                 result = DeleteDomain(domain_name);
37             }

```

```
38     else if (operation == "register")
39     {
40         if (args.Length < 2)
41         {
42             Runtime.Log("Required arguments: [\"domain_name\", \"owner\"]");
43             result = 0;
44         }
45         else
46         {
47             byte[] owner = (byte[])args[1];
48             result = RegisterDomain(domain_name, owner);
49         }
50     }
51     else if (operation == "transfer")
52     {
53         if (args.Length < 2)
54         {
55             Runtime.Log("Required arguments: [\"domain_name\", \"to_address\"]");
56             result = 0;
57         }
58         else
59         {
60             byte[] to_address = (byte[])args[1];
61             result = TransferDomain(domain_name, to_address);
62         }
63     }
64     else
65     {
66         result = false;
67     }
68 }
69
70     return result;
71 }
```

lab5_domain.cs* X

C# lab5_domain lab5_domain.lab5_domain

```
73     private static object QueryDomain(string domain_name)
74     {
75         object result = 0;
76
77         Runtime.Notify("QueryDomain", domain_name);
78
79         StorageContext ctx = Storage.CurrentContext;
80         byte[] owner = Storage.Get(ctx, domain_name);
81         if (owner.Length == 0)
82         {
83             Runtime.Notify("Domain is not registered", domain_name);
84             result = 0;
85         }
86         else
87         {
88             result = owner;
89         }
90
91         return result;
92     }
```



```
94     private static bool RegisterDomain(string domain_name, byte[] owner)
95     {
96         bool result = false;
97
98         Runtime.Notify("RegisterDomain", domain_name, owner);
99
100        if (!Runtime.CheckWitness(owner))
101        {
102            Runtime.Notify("Owner argument is not the same as the sender", owner);
103            result = false;
104        }
105        else
106        {
107            StorageContext ctx = Storage.CurrentContext;
108            byte[] exists = Storage.Get(ctx, domain_name);
109            if (exists.Length != 0)
110            {
111                Runtime.Notify("Domain is already registered", domain_name, exists);
112                result = false;
113            }
114            else
115            {
116                Storage.Put(ctx, domain_name, owner);
117                result = true;
118            }
119        }
120    }
121    return result;
122 }
```

```
124     private static bool TransferDomain(string domain_name, byte[] to_address)
125     {
126         bool result = false;
127
128         Runtime.Notify("TransferDomain", domain_name, to_address);
129
130         StorageContext ctx = Storage.CurrentContext;
131         byte[] owner = Storage.Get(ctx, domain_name);
132         if (owner.Length == 0)
133         {
134             Runtime.Notify("Domain is not registered", domain_name);
135             result = false;
136         }
137         else
138         {
139             if (!Runtime.CheckWitness(owner))
140             {
141                 Runtime.Notify("Sender is not the owner on this domain, cannot transfer", domain_name);
142                 result = false;
143             }
144             else
145             {
146                 if (to_address.Length != 34)
147                 {
148                     Runtime.Notify("Invalid new owner address. Must be exactly 34 characters", to_address);
149                     result = false;
150                 }
151                 else
152                 {
153                     Storage.Put(ctx, domain_name, to_address);
154                     result = true;
155                 }
156             }
157         }
158
159         return result;
160     }
```

```
124     private static bool TransferDomain(string domain_name, byte[] to_address)
125     {
126         bool result = false;
127
128         Runtime.Notify("TransferDomain", domain_name, to_address);
129
130         StorageContext ctx = Storage.CurrentContext;
131         byte[] owner = Storage.Get(ctx, domain_name);
132         if (owner.Length == 0)
133         {
134             Runtime.Notify("Domain is not registered", domain_name);
135             result = false;
136         }
137         else
138         {
139             if (!Runtime.CheckWitness(owner))
140             {
141                 Runtime.Notify("Sender is not the owner on this domain, cannot transfer", domain_name);
142                 result = false;
143             }
144             else
145             {
146                 if (to_address.Length != 34)
147                 {
148                     Runtime.Notify("Invalid new owner address. Must be exactly 34 characters", to_address);
149                     result = false;
150                 }
151                 else
152                 {
153                     Storage.Put(ctx, domain_name, to_address);
154                     result = true;
155                 }
156             }
157         }
158
159         return result;
160     }
```

```
162     private static bool DeleteDomain(string domain_name)
163     {
164         bool result = false;
165
166         Runtime.Notify("DeleteDomain", domain_name);
167
168         StorageContext ctx = Storage.CurrentContext;
169         byte[] owner = Storage.Get(ctx, domain_name);
170         if (owner.Length == 0)
171         {
172             Runtime.Notify("Domain is not registered", domain_name);
173             result = false;
174         }
175         else
176         {
177             if (!Runtime.CheckWitness(owner))
178             {
179                 Runtime.Notify("Sender is not the owner on this domain, cannot delete", domain_name);
180                 result = false;
181             }
182             else
183             {
184                 Storage.Delete(ctx, domain_name);
185                 result = true;
186             }
187         }
188
189         return result;
190     }
191 }
```

Lab 6 - lab6_NEP5pattern.cs

NEO Smart Contract Patterns

lab6_NEP5pattern.cs

C# lab6_NEP5pattern

lab6_NEP5pattern.NEP5Base

TotalSupply

```
1  using Neo.SmartContract.Framework;
2  using Neo.SmartContract.Framework.Services.Neo;
3  using System;
4  using System.Numerics;
5
6  // NEP5 Token Proposal: https://github.com/neo-project/proposals/blob/master/nep-5.mediawiki
7
8  namespace lab6_NEP5pattern
9  {
10     public class NEP5Base
11     {
12         public BigInteger TotalSupply;
13         public string Name;
14         public string Symbol;
15         public byte Decimals;
16         public byte[] OwnerAccountScriptHash;
17     }
18
19     public class lab6_NEP5pattern : SmartContract
20     {
21         static readonly byte[] _OwnerAccountScriptHash = "ATrzHaicmhRj15C3Vv6e6gLfLqhSD2PtTr".ToScriptHash();
22
23         public static event Action<byte[], byte[], BigInteger> transfer;
24
25         public static object Main(string operation, params object[] args)
26         {
27             object result = false; // = 0 (zero)
28
29             NEP5Base TOKENBASE = new NEP5Base { TotalSupply = 10000,
30                                              Name = "My Test Token",
31                                              Symbol = "MTT",
32                                              Decimals = 8,
33                                              OwnerAccountScriptHash = _OwnerAccountScriptHash };
```

lab6_NEP5pattern.cs X

C# lab6_NEP5pattern lab6_NEP5pattern.NEP5Base TotalSupply

```
35     if (operation == "totalSupply")
36     {
37         Runtime.Notify("totalSupply");
38         result = TotalSupply(TOKENBASE);
39     }
40     else if (operation == "name")
41     {
42         Runtime.Notify("name");
43         result = Name(TOKENBASE);
44     }
45     else if (operation == "symbol")
46     {
47         Runtime.Notify("symbol");
48         result = Symbol(TOKENBASE);
49     }
50     else if (operation == "decimals")
51     {
52         Runtime.Notify("decimals");
53         result = Decimals(TOKENBASE);
54     }
55     else if (operation == "balanceOf")
```

```
55     else if (operation == "balanceOf")
56     {
57         if (args.Length < 1)
58         {
59             result = false;
60         }
61         else
62         {
63             byte[] account = (byte[])args[0];
64             Runtime.Notify("balanceOf");
65             result = BalanceOf(TOKENBASE, account);
66         }
67     }
68     else if (operation == "transfer")
69     {
70         if (args.Length < 3)
71         {
72             result = false;
73         }
74         else
75         {
76             byte[] from = (byte[])args[0];
77             byte[] to = (byte[])args[1];
78             BigInteger amount = (BigInteger)args[2];
79             Runtime.Notify("transfer", args[0], args[1], args[2]);
80             result = Transfer(TOKENBASE, from, to, amount);
81         }
82     }
```

```
83         else if (operation == "deploy")
84     {
85         Runtime.Notify("deploy");
86         result = Deploy(TOKENBASE);
87     }
88     else
89     {
90         result = false;
91     }
92
93     return result;
94 }
95
96     private static BigInteger TotalSupply(NEP5Base TOKENBASE)
97     {
98         return TOKENBASE.TotalSupply;
99     }
100
101    private static string Name(NEP5Base TOKENBASE)
102    {
103        return TOKENBASE.Name;
104    }
105
106    private static string Symbol(NEP5Base TOKENBASE)
107    {
108        return TOKENBASE.Symbol;
109    }
110
111    private static byte Decimals(NEP5Base TOKENBASE)
112    {
113        return TOKENBASE.Decimals;
114    }
```

```
116     private static BigInteger BalanceOf(NEP5Base TOKENBASE, byte[] account)
117     {
118         BigInteger result = 0;
119
120         StorageContext ctx = Storage.CurrentContext;
121         BigInteger currentBalance = Storage.Get(ctx, account).AsBigInteger();
122
123         result = currentBalance;
124
125         return result;
126     }
127
128     private static bool Deploy(NEP5Base TOKENBASE)
129     {
130         bool result = false;
131
132         if (Runtime.CheckWitness(TOKENBASE.OwnerAccountScriptHash))
133         {
134             StorageContext ctx = Storage.CurrentContext;
135
136             // Create on-chain ledger entry for the owner of this token. Check to see if the ledger already
137             byte[] currentBalance = Storage.Get(ctx, TOKENBASE.OwnerAccountScriptHash);
138             if (currentBalance.Length == 0)
139             {
140                 Storage.Put(ctx, TOKENBASE.OwnerAccountScriptHash, TOKENBASE.TotalSupply);
141                 result = true;
142             }
143         }
144
145         return result;
146     }
```

```
148     private static bool Transfer(NEP5Base TOKENBASE, byte[] from, byte[] to, BigInteger amount)
149     {
150         bool result = false;
151
152         if (amount > 0)
153         {
154             if (Runtime.CheckWitness(from)) // is the account invoking this contract == "from" account
155             {
156                 if (from == to)
157                 {
158                     result = true;
159                 }
160                 else // from != to
161                 {
162                     StorageContext ctx = Storage.CurrentContext;
163
164                     // Get balance from the "from" ledger
165                     BigInteger fromBalance = Storage.Get(ctx, from).AsBigInteger();
166                     if (fromBalance > amount)
167                     {
168                         // Update "from" ledger
169                         Storage.Put(ctx, from, fromBalance - amount);
170
171                         // Update "to" ledger
172                         BigInteger toBalance = Storage.Get(ctx, to).AsBigInteger();
173                         Storage.Put(ctx, to, toBalance + amount);
174
175                         // Log a "transfer" event
176                         transfer(from, to, amount);
177
178                         result = true;
179                     }
180                 }
181             }
182         }
183
184         return result;
185     }
```

End of Module 2

- ▶ Lab 4 - lab6_processoperationpattern.cs
- ▶ Lab 5 - lab5_domain.cs
- ▶ Lab 6 - lab6_NEP5pattern.cs

4. Why NEO?

NEO was created for Large Scale dApps*

No transaction fees
Initial 10 GAS fee is
for free. Simple
Smart Contracts can
be run for free.

Mainchain speeds
NEO 1000 - 10,000 TPS (single core)
BTC 3 - 4 TPS
ETH 10 - 20 TPS

Confirmations needed
NEO 1 and only 1
BTC 6+
ETH 20+

More freedom for
application developers
to choose the business
model that most suits to
their business. Service
and usage fees can be
charged in any way off-
chain.

*dApps - Distributed Apps

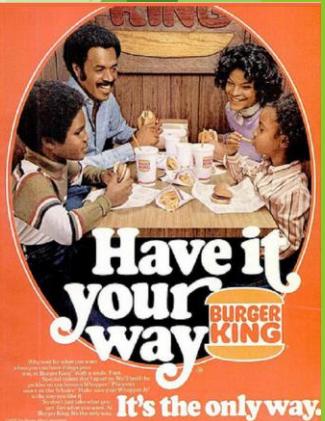
NEO C# NET Developers Center of Excellence

NEO Persistable Classes make Enterprise App Development real

5. 3rd Generation Distributed App (dApp) Platform

Smart Contracts: “Have it your Way”

- ▶ Develop, debug and test Smart Contracts in the language you prefer
 - ▶ C# (reference implementation)
 - ▶ Python, Java, Golang, JavaScript
- ▶ Develop, debug and test Client Apps using the frameworks you prefer
 - ▶ ASP.NET, WinForms, WPF, Console Applications, Window Services, ...
 - ▶ Node.JS
 - ▶ Python, Java, Golang, JavaScript



NEO C# Developer Tool Suite (March 2018)

- ▶ NEO Blockchain Quick Start Guide for .NET Developers (mwherman2000/neo-dotnetquickstart)
- ▶ NEO Persistable Classes (NPC) Compiler (mwherman/neo-npcc)
- ▶ NEO NEON Compiler (neo-project/neo-compiler)
- ▶ NEO Debugger (CityOfZion/neo-debugger-tools)
- ▶ NEO Node/Wallet (Developer Edition) (CityOfZion/neo-gui-developer)
- ▶ NEO PrivateNet Docker Container (metachris/neo-privnet-with-gas)
- ▶ NEO TestNet Network (http://docs.neo.org/en-us/node/testnet.html)
- ▶ NEO MainNet Network

Smart Contracts: “Have it your way”

```
using Neo.SmartContract.Framework;
using Neo.SmartContract.Framework.Services.Neo;

namespace Neo.SmartContract
{
    public class Lock : VerificationCode
    {
        public static bool Verify(byte[] signature)
        {
            Header header = Blockchain.GetHeader(Blockchain.GetHeight());
            if (header.Timestamp < 1499328600) // 2017-6-6 18:10
                return false;
            // Paste the public key byte array here
            return VerifySignature(
4, 91, 78, 93, 139, 126, 48, 58,
2, 122, 213, 170 }, signature);
        }
    }
}

import org.neo.smartcontract.framework.SmartContract;
public class ContractAuthentication extends SmartContract {
    static boolean Main(byte[] signature)
    {
        Header header = Blockchain.getHeader(Blockchain.height());
        if (header.timestamp() < 1506933900)
            return false;
        // Paste the public key byte array here
        byte[] b = { 2, -99, 6, 102, 4, -41, 48, -96, -116, 23, 9, 72, -89, -104, -107, 2, -8, -70, -2,
                    96, 60, -21, 105, 105, -93, 103, -80, -113, 17, -61, 7, 20, -85 };
        return verifySignature(b, signature);
    }
}
```

Smart Contract Development: NEO or ETH?

Compare/contrast Ethereum and the NEO Blockchain

Ethereum

- ▶ ETH Smart contract == single object/entity with a simple set of methods and simple types of data
- ▶ The Ideal Application?
 - ▶ Blockchain version 2.0 Alt tokens
 - ▶ Simple, single entity with a few methods
- ▶ Beyond that?
 - ▶ Very little future for Ethereum as an Enterprise Application platform
 - ▶ “One trick pony” x 1500

NEO

- ▶ NEO Smart contract == a program capable of working with dozens of different types of objects/entities
- ▶ Unlimited potential as an Enterprise Application Platform
- ▶ Will drive acquisition of NEO
- ▶ Automatic accumulation of GAS
- ▶ Increased ability to participate in the NEO Smart Economy
 - ▶ Voting as well as having operating funds

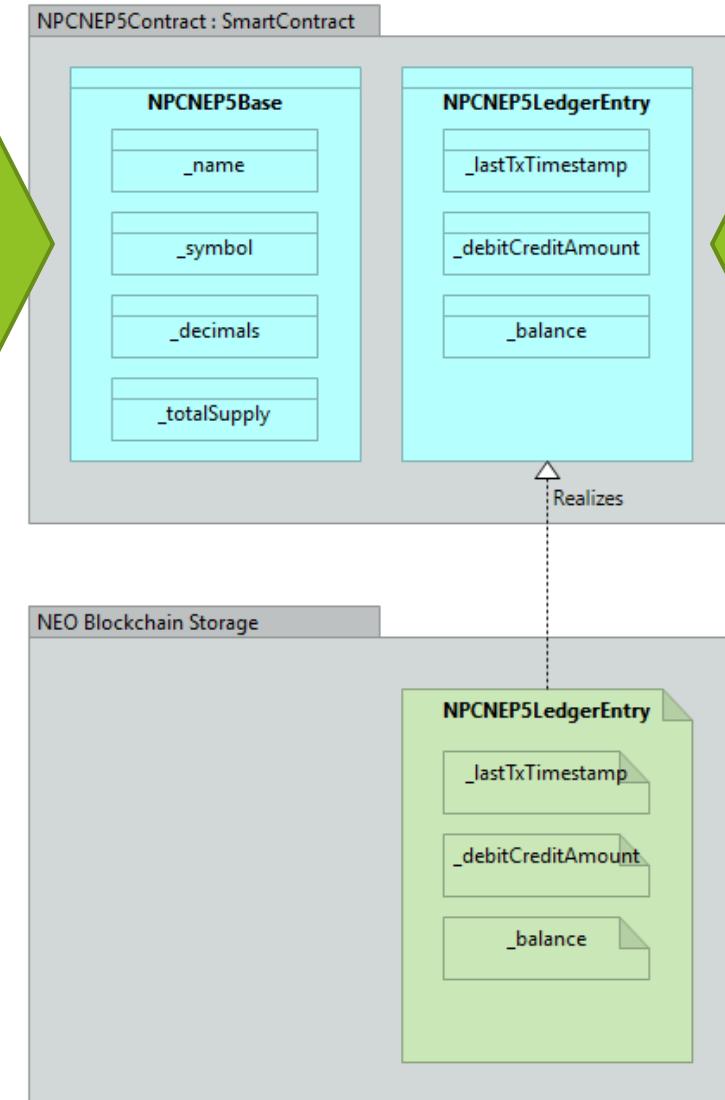
NEO Persistable Classes (NPC) NEP5 Token

Reference Implementation

```
namespace NPC.mwherman2000.NPCNEP5Token1.Model
{
    public interface NPCLevel0Basic { }
    public interface NPCLevel1Managed { }
    public interface NPCLevel2Persistable { }
    public interface NPCLevel3Deletable { }
    public interface NPCLevel4Collectible { }
    public interface NPCLevel4CollectibleExt { }

    public interface NPCLevel0CustomMethods { }
    public interface NPCLevel1CustomMethods { }
    public interface NPCLevel2CustomMethods { }
    public interface NPCLevel3CustomMethods { }
    public interface NPCLevel4CustomMethods { }

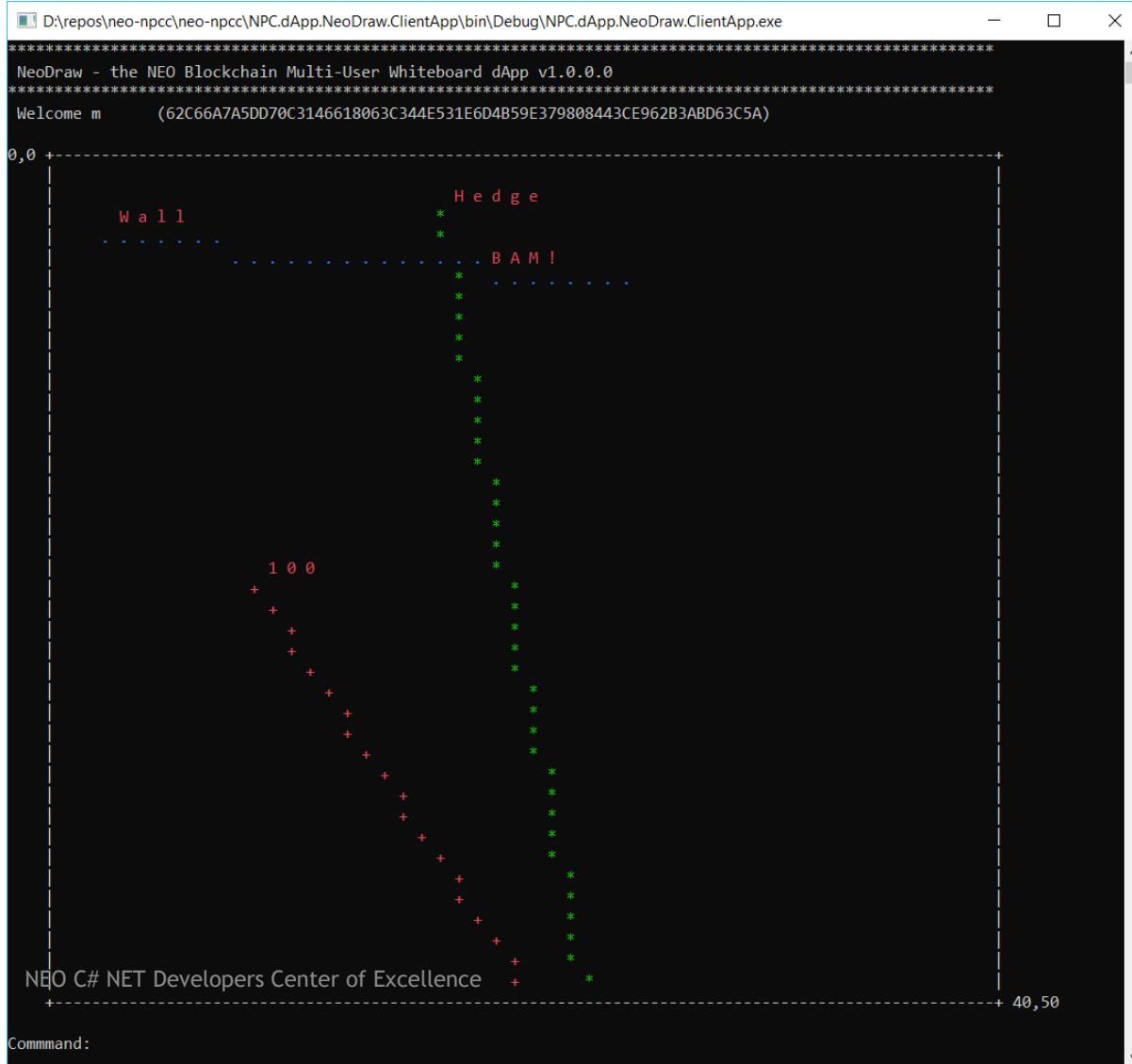
    public class NPCNEP5Base : NPCLevel0Basic,
                                NPCLevel1Managed
    {
        public string name;
        public string symbol;
        public Int32 decimals;
        public BigInteger totalSupply;
    }
}
```



6. Smart Contract Development using Entities

NeoDraw - NEO Multi-Person Shared Whiteboard dApp

NEO Blockchain as an Enterprise Application Platform



- ▶ NeoDraw is advanced proof-of-concept distributed application (dApp) for the NEO Persistable Classes (NPC) Entity-based dApp (e-dApp) Platform running on the NEO Blockchain.
- ▶ Uses 3 entities
 - ▶ UserCredentials
 - ▶ User name and User Password
 - ▶ UserPoint
 - ▶ Array of (X,Y) coordinates/user
 - ▶ NeoCounter
 - ▶ Tracks number points/user

NeoDraw - NEO Multi-Person Shared Whiteboard dApp

NEO Blockchain as an Enterprise Application Platform

```
UserCredentials.cs  ✎ X  UserPoint.cs  ✎ X  NeoCounter.cs  ✎ X
C# NPC.dApps.NeoDraw  C# NPC.dApps.NeoDraw  C# NPC.dApps.NeoDraw  ▾ C# NPC.dApps.NeoDraw.NeoCounter
1  namespace NPC.dApps.NeoDraw
2  {
3      public class UserCredentials
4      {
5          public string Username { get; set; }
6          public string Password { get; set; }
7          public string Email { get; set; }
8          public string Nickname { get; set; }
9          public string Bio { get; set; }
10         public string Avatar { get; set; }
11         public string LastSeen { get; set; }
12         public string LastActivity { get; set; }
13         public string LastLocation { get; set; }
14         public string LastDevice { get; set; }
15     }
16 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Numerics;
7
8  namespace NPC.dApps.NeoDraw
9  {
10     public partial class NeoCounter : NPCLevel0Basic,
11                                 NPCLevel1Managed,
12                                 NPCLevel2Persistable,
13                                 NPCLevel3Deleteable,
14                                 NPCLevel4Collectible,
15                                 NPCLevel4CollectibleExt
16     {
17         public BigInteger currentNumber; // Next number to give out
18     }
19 }
```

```
20
21
22     {
23         public byte[] encodedUsername;
24         public byte[] encodedPassword;
25     }
26 }
```

NEO C# .NET Developers Center of Excellence 4/1/2018 80

NPC Layered Functionality

Level 0 Basic

- ▶ Basic class definition with no supporting methods

Level 1 Managed

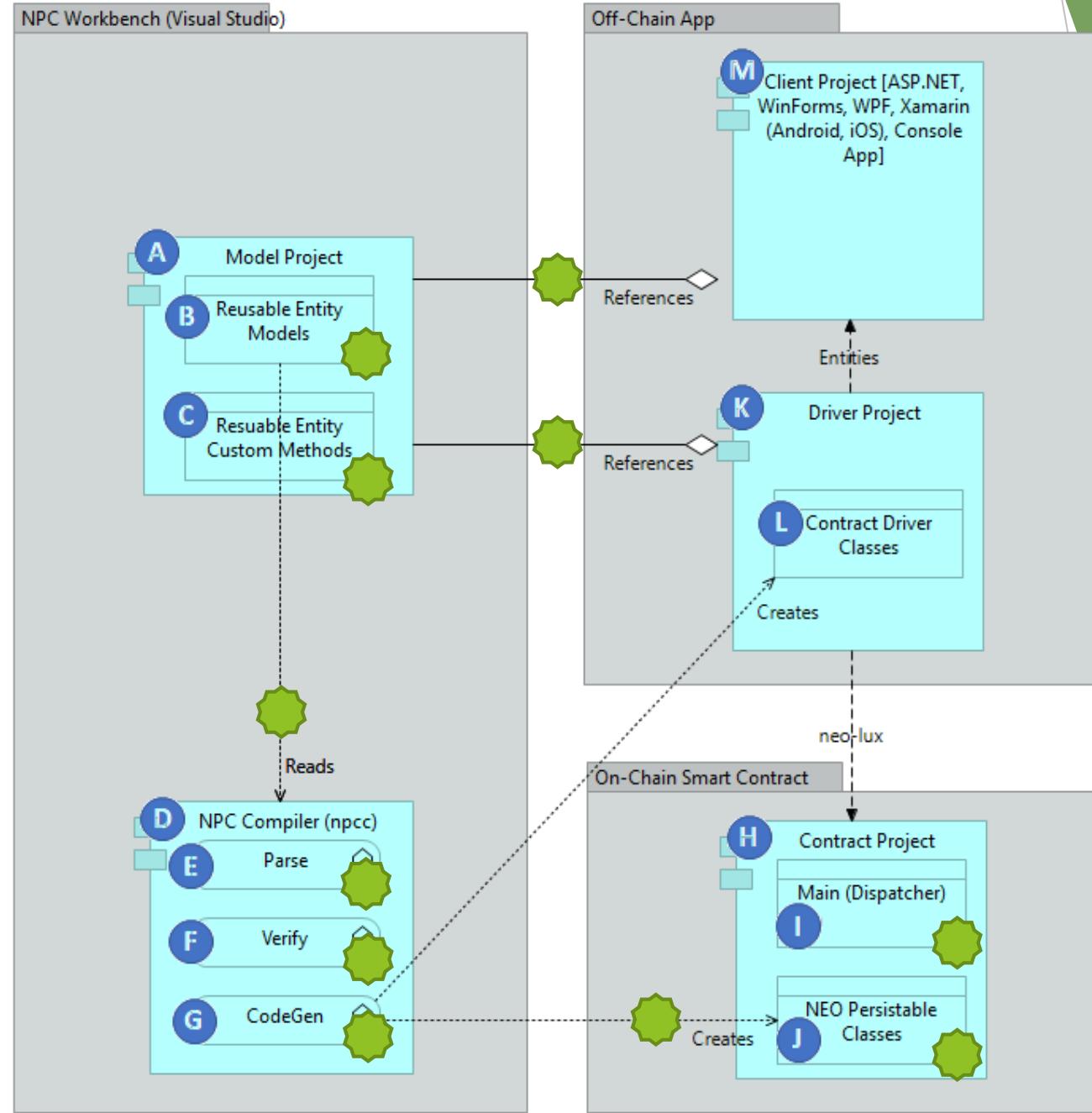
- ▶ `New()`
- ▶ `New(...)`
- ▶ `Null()`
- ▶ `IsNull(e)`
- ▶ `Set(e, ...)`
- ▶ `Setfieldname(e, value)`
- ▶ `Getfieldname(e)`
- ▶ `Log("label", e)`

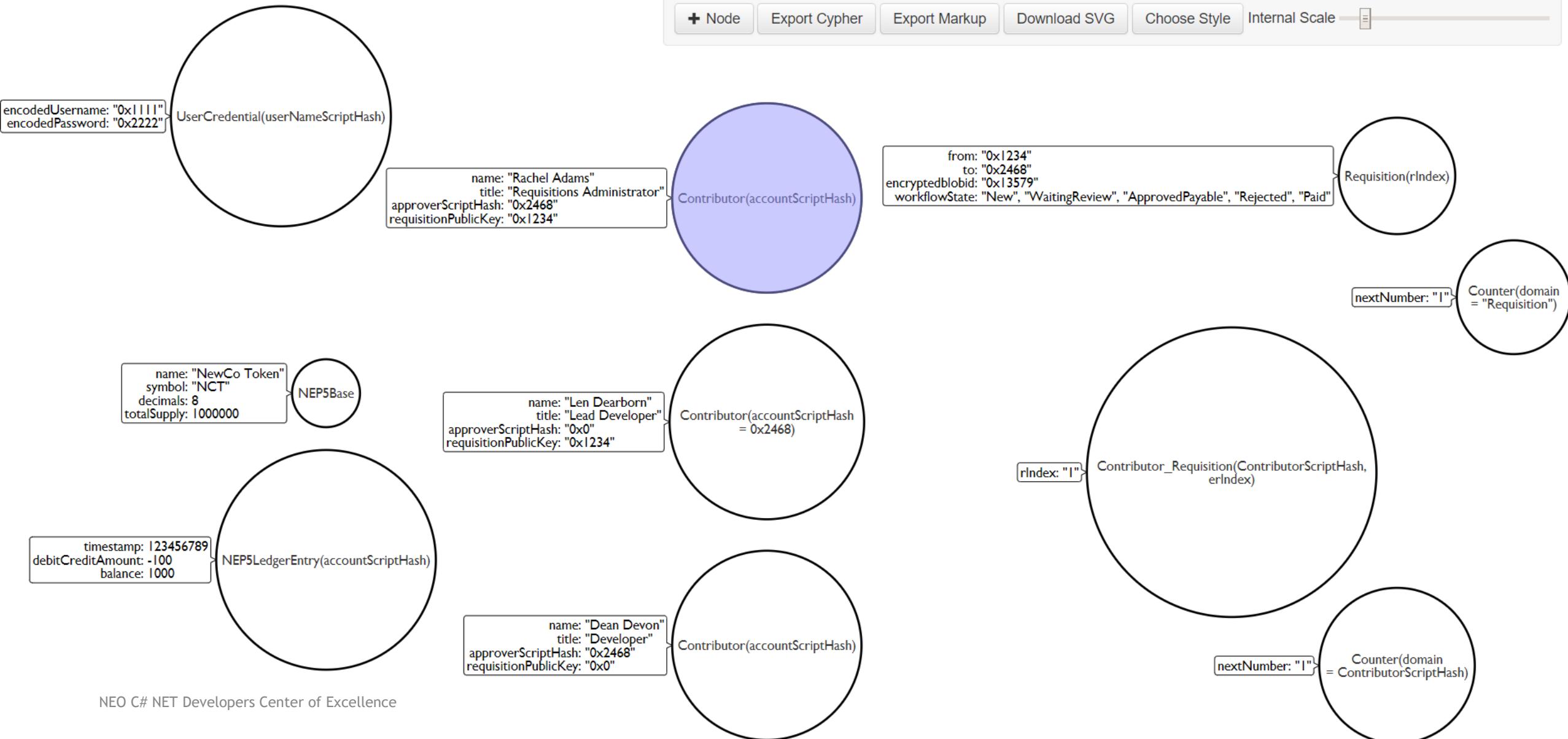
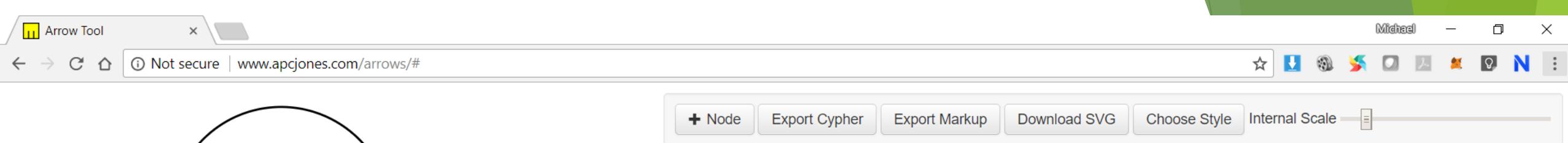
Level 2 Persistable

- ▶ `Missing()`
- ▶ `IsMissing(e)`
- ▶ `Put(e, string key)`
- ▶ `Put(e, byte[] key)`
- ▶ `Get(string key)`
- ▶ `Get(byte[] key)`

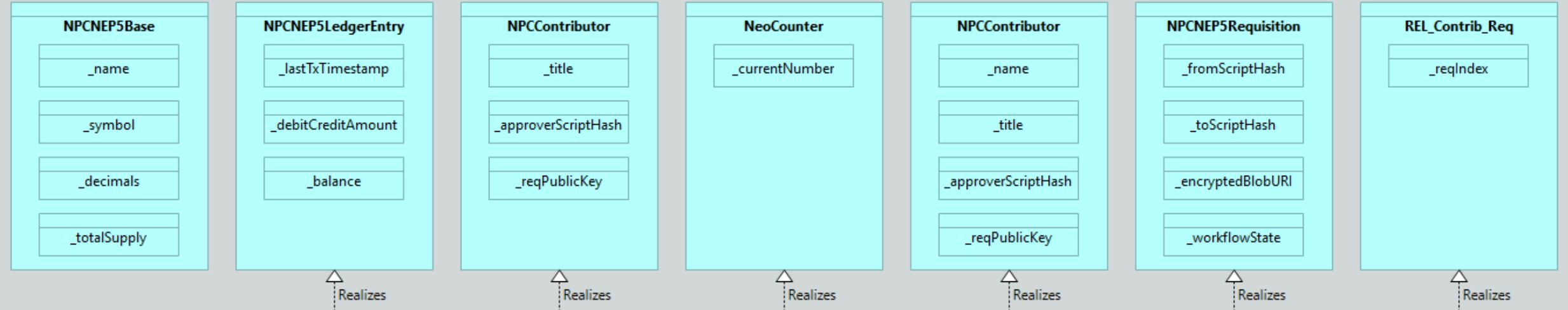
Architecture

NEO Persistable Classes (NPC)
Platform 2.1 Architecture

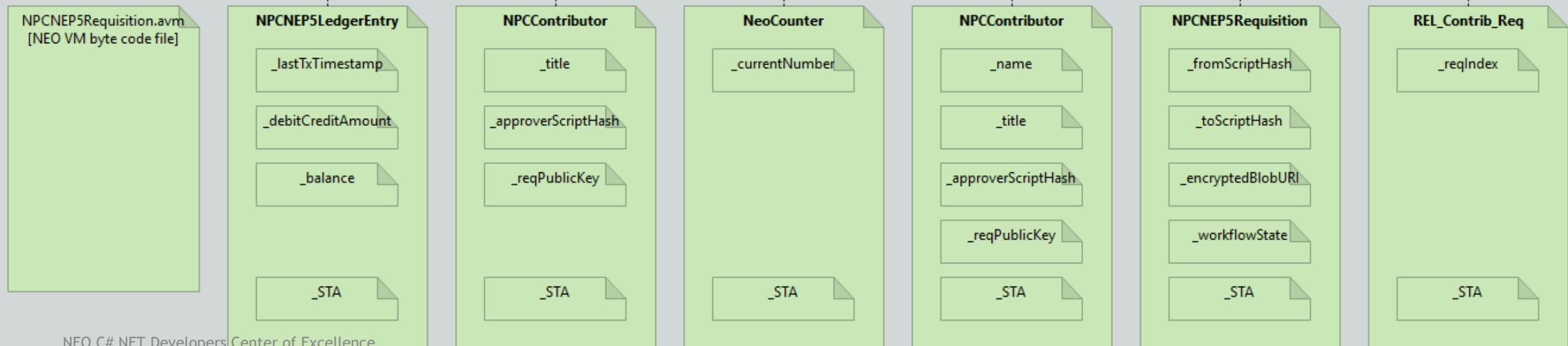




NPCNEP5Contract : SmartContract



NEO Blockchain Storage



7. NEO Evidence

NEO Events



Second Wednesday evening of each month



NEO C# .NET Developers Center of Excellence

HackCU Episode IV · University of Colorado Boulder
600 Hackers · 24 Hours · February 24th and 25th 2018



4/1/2018

NEO Events (con't)



Second Thursday evening of each month



1 Week \$NEO Wallet Design Challenge!

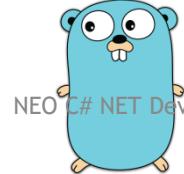
I will select 3 designs to each win 5 GAS.

- 1) Create a UI & UX Design for a slick NEO Wallet for desktop.
- 2) Email your submission to challenge@dean.press
- 3) Submission deadline is March 7th.
- 4) Winners announced within a few days.

Good luck! 🎉



NEO dApp Competitions ~USD\$500K in prizes



NEO C# .NET Developers Center of Excellence

1st NEO Dev Competition

Nov. 20, 2017 - Mar. 31, 2018



The background features a dark, abstract network graph with glowing nodes and lines, symbolizing a decentralized or interconnected system.

 **NEO**
smart economy

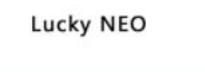
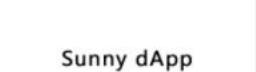
 Microsoft

CITY OF ZION IS LAUNCHING ITS SECOND

DAPPS COMPETITION

And we want you to get involved

The City of Zion council is promoting a decentralised apps (dApps) competition to further elevate the NEO developer community.

 NEX by The NEX Team NEX combines the NEO blockchain with an off-chain matching engine to enable much faster and more complex trades than existing decentralized exchanges. Website Whitepaper Twitter LinkedIn Medium	 Moonlight by The Moonlight Team Moonlight is a distributed workforce and analytical project management platform featuring a global public ledger of contributor work experience and a new match-making algorithm to effectively fulfill project needs. Website	 Red Pulse by Red Pulse Red Pulse Tokens (RPX) are NEO tokens issued by Red Pulse, an event-driven research firm covering market events impacting Chinese companies, sectors and the overall economy. Website Reddit Contract	 AdEx by adexnetwork AdEx is a decentralized ad exchange built on the blockchain and smart contracts. The core feature of AdEx will be the so-called AdEx User Profile - a personalized page that allows every end user to understand and control the ads delivered to them. Website Medium Facebook Telegram Twitter Reddit	 NeoAuth by @NeoAuth NeoAuth enables authentication over the NEO blockchain, allowing you to log in with a NEO address instead of an email and password. Website Demo GitHub Whitepaper	 ZEEPIN by ZEEPIN The Distributed Creative New Economy. ZEEPIN, a decentralized innovation community, is dedicated to promoting highly efficient circulation of innovation assets. Smart work. Creative life! Website Telegram Twitter Facebook Contract	 Qlink by Alkin Li Qlink, developed by Qlink Foundation in Singapore, adopts the blockchain technology and creates a decentralized mobile network for P2Peer WiFi sharing, mobile data converted content distribution, enterprise telecom services and crowd-sourcing base stations (including Medium Page AMA summary Contract	 Neo Smart IoT by halo2328, phattar Control IoT (Internet of Things) devices via Neo smart contracts (first device is an ESP8266). Website GitHub Contract	 imusify by DavidWalters123, geek96, Nikolaj K, metachris imusify is a free, blockchain based, incentivized and decentralized platform for music related digital content such as audio, video, apps, images, and blogging where anyone can join, contribute and get paid \$IMU. Website GitHub
 Chain Line by ruitatechstar Peer-to-peer courier platform. Couriers transport items to fulfill demands and earn courier fees. Website GitHub	 Phantasma by Rytis Phantasma is a platform where the users control their own content, instead of relying in third parties servers. The platform support any kind of transactions between users, e.g. email, chat, files, money transfers. NEO was used for this project due to its fast transactions and C# Demo Website GitHub	 NeoTrade by barmas NeoTrade is a NEP5 trading platform. Inspired by EtherDelta, NeoTrade is a NEO based smart contract that allows users to deposit NEO, GAS or any other NEP-5 asset directly to the smart contract and trade amongst themselves. Website GitHub	 Turing Complete Smart Contract by Nikita K A smart contract which interprets encoded classical Turing machines. Youtube GitHub	 KRYPTON.live by MediaServe KRYPTON is a Smart Contract and ecosystem to provide SIP registration on the NEO blockchain using a KRYPTON compatible SIP provider. Youtube Demo GitHub	 Switcheo by RovenKee, hennyhua, jacyuu Switcheo is a decentralised exchange built on NEO's blockchain. It supports trading of NEP-5 tokens and NEO system assets like GAS & NEO. It aims to be a truly decentralised exchange while still providing for a super user experience. Youtube Website GitHub	 Trip Shares by xtalia Trip sharing with deposits. Travellers commit to a shared seat by depositing \$TRS NEP-5 tokens, which are used as insurance when the passenger cancels the trip after a set date. Website GitHub	 BlockSaver by impravice, monica Saving smart contract, with interests and fines. You can define how long to lock NEO and GAS assets, the smart contract defines an interest rate, and the system has a penalty for early withdrawal. GitHub	 Lucky NEO by miroslavc, licorng, StefanDobach, WallyLeung, tammerman Lucky NEO allows anyone to send their extra gas to a raffle contest. One winner will be chosen every two weeks and automatically paid out. Lucky NEO uses an admin account to automatically pay out the winner to the address that sent the funds; so you don't have to follow Post GitHub
 Neo Fund	 Neo Raffle by amarthu A smart contract that enables a raffle/lottery on the Neo block chain. Send GAS, win GAS! This project also demonstrates a solid way to use random numbers in a smart contract GitHub	 Nep Swap by block_sharan Protocol and smart contract for trading NEP-5 based assets.NepSwap is a protocol and smart contract for trading NEP-5 assets. This prototype idea is based on Etherdelta and Bancor. It doesn't have a personal token or ICO, this project is intended for community, and Github	 Smart Promise by SergeyAntonenko, marylonya Smart Promise is an electronic journal of smart promises developed through Blockchain. User of such an environment receives a reward for each of his fulfilled promises. The concept of the idea contains the possibility to get a strong motivation to action that user pointed in Youtube Website GitHub Whitepaper	 Sunny dApp by Jonrhundrederberg This dApp allows you to insure against bad weather conditions on a given day. If the relative sunshine duration on that day is lower than 50 percent, you get paid. Website GitHub	 NEO Name Service by NewEconoLab(NEL) Distributed domain name service ending with .neo based on NEO blockchain Website GitHub	 THEKEY by THEKEY team THEKEY is a Decentralized Ecosystem of Identity Verification Tool Using National Big-data and Blockchain. THEKEY Project Team is now developing an identification verification (IDV) tool with blockchain based dynamic multi-dimension Website GitHub telegram facebook twitter Contract	 蓝鲸淘 by 中·科技 蓝鲸淘智能资产去中心化管理平台，通过开源的区块链技术与智能合约规则，为用户提供点对点、无中介的数字资产流转服务，降低数字资产流转过程中的使用成本。 Website GitHub	 TRINITY by Trinity team Trinity is applicable to blockchain transfer of NEP-5 standard tokens. As a Neo-version lightning network, Trinity achieves real-time performance, low transaction fees, scalability, and privacy protection of Neo mainnet assets through state channels website GitHub telegram twitter Contract
 GAME.FUND	 CONCIERGE Developers Center of Excellence	 Damn Single	 AISC AI SMART CONTRACT	 ORBIS	 NARRATIVE	 Aphelion	 Blockchain Store	 Bridge Protocol

Where We Are: Weiss Ratings

Press Esc to exit full screen

A: Excellent
(0)



B: Good
(5)

- Ethereum
- EOS
- **NEO**
- Cardano
- Steem

C: Fair
(59)

- Bitcoin
- Bitcoin Cash
- Ripple
- Litecoin
- Qtum

D: Weak
(18)

- Bitcoin Gold
- Metaverse

Weiss
Ratings

Independent. Unbiased. Accurate. Trusted.

Risk - price volatility**Reward** - return fluctuations**Technology** - infrastructure and flexibility**Fundamentals** - currency specific factors, market, and public support

NEO C# NET Developers Center of Excellence



NEO News: Switchers

- ▶ [QTUM to NEO] THEKEY, a Decentralized Ecosystem of Identity Verification Tool Using National Big-data and [NEO] Blockchain
https://www.reddit.com/r/NEO/comments/7areac/ama_on_9th_nov_thekey_a_decentralized_ecosystem/
- ▶ [ETH to NEO] Guardian Circle announce switch to NEO platform
<https://neonewstoday.com/general/guardium-announce-switch-to-neo-platform/>
“...the more we dug into NEO, the more we realized that they had really great answers to every one of our concerns with Ethereum and many additional benefits that are unique to the NEO ecosystem.”
- ▶ [ETH to NEO] Narrative project: Big News: KYC and Our Switch to NEO
<https://blog.narrative.network/big-news-kyc-and-our-switch-to-neo-2f34215beef9>
- ▶ [ETH to NEO] Concierge.io NEO Token Sale
<https://medium.com/@concierge.io/concierge-io-neo-token-sale-f870a0a27b04>
- ▶ [ETH to NEO] IMPORTANT NEWS FOR THE WOWOO COMMUNITY
http://wooonet.com/news.html#news_2018_6
- ▶ [ETH to NEO] Three things to consider when choosing a blockchain for your project—or why Veris chose NEO.
<https://medium.com/verisfoundation/three-things-to-consider-when-choosing-a-blockchain-for-your-project-or-why-veris-chose-neo-b4483135c382>
- ▶ [Blockchain agnostic strategy] Jibrel Network founder looking to switch from Ethereum to NEO
<https://imgur.com/oHYTYi2>



NARRATIVE

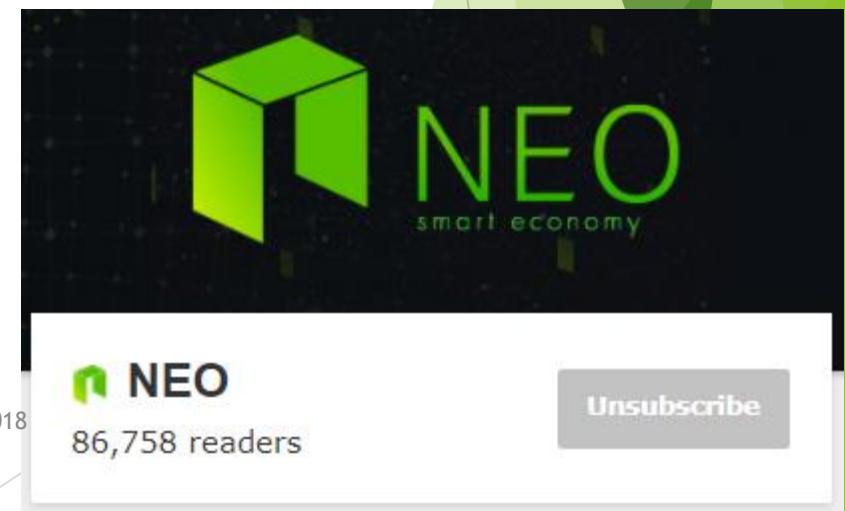


Wowoo

8. Resources

NEO Resources: Videos and Webcasts

- ▶ The New Vision - Da Hongfei
<https://www.youtube.com/watch?v=th7jZlmoZBc>
- ▶ Malcolm Lerider: Clarification on NEO, GAS and Consensus Nodes
<https://www.youtube.com/watch?v=iV4WREWtCil>
- ▶ A Deep Dive into NeoVM & NeoContract [English Subtitles]
<https://www.youtube.com/watch?v=fLppte-guYE>
- ▶ NEO Persistable Classes (NPC) e-dApp Smart Contract Platform 2.0: Deep Dive
<https://www.youtube.com/watch?v=Nj4-m2o94VE>
- ▶ NEO - Reddit
<https://www.reddit.com/r/NEO/>



NEO Resources: NEO DevCon 1, January 2018

- ▶ The New Vision - Da Hongfei | NEO DevCon 1
<https://www.youtube.com/watch?v=th7jZlmoZBc>
- ▶ Opening Address - Scott Hunter, Microsoft | NEO DevCon 1
<https://www.youtube.com/watch?v=ZFsu5HPU2xM>
- ▶ Ontology, The Technical Vision of Distributed Trust Networks | NEO DevCon 1
<https://www.youtube.com/watch?v=QyaZz0vtONs>
- ▶ Imusify, Decentralized Platform for Music Related Digital Content | NEO DevCon 1
<https://www.youtube.com/watch?v=dgmfO8nKJkE>
- ▶ NeoAuth, Login Using the NEO Blockchain | NEO DevCon 1
<https://www.youtube.com/watch?v=F7teukbcblg>
- ▶ Phantasma, User-Owned Content | NEO DevCon 1
<https://www.youtube.com/watch?v=cinexz6gjsU>
- ▶ NEO•ONE, Development Framework | NEO DevCon 1
<https://www.youtube.com/watch?v=yWzjNhiZFmA>
- ▶ THEKEY, A Decentralized Ecosystem of An Identity Verification Tool | NEO DevCon 1
<https://www.youtube.com/watch?v=St-UW0lPB7o>
- ▶ NEO's Global Vision and the Evolving Regulatory Environment - Miles Graham | NEO DevCon 1
<https://www.youtube.com/watch?v=FidRGNeab3I>
- ▶ NEX, Decentralized Exchanges | NEO DevCon 1
<https://www.youtube.com/watch?v=1005ed20e0Q>

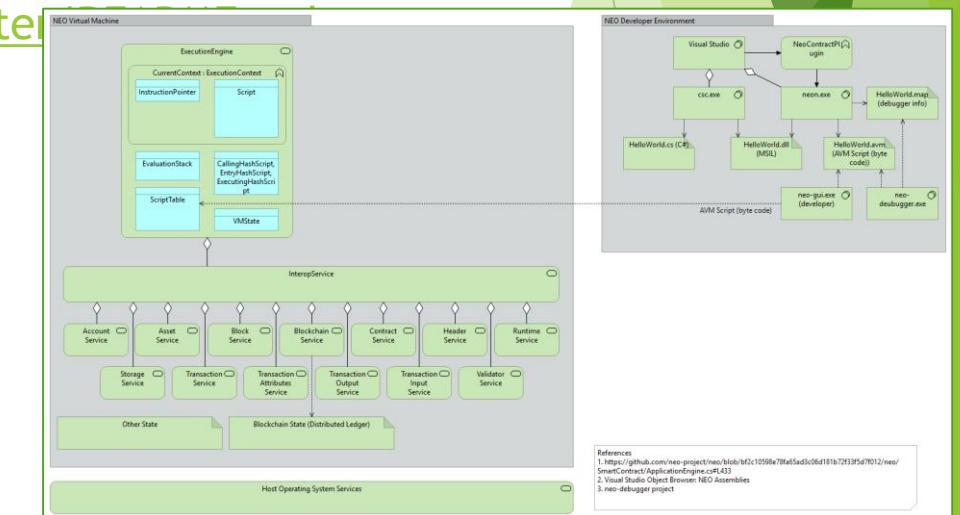
NEO Resources

General

- ▶ NEO.org
<http://neo.org>
- ▶ NEO Tutorials (C#, Java, Python), API docs
<http://docs.neo.org>
- ▶ NEO DevCon “The New Vision” Keynote by NEO Founder
https://www.youtube.com/watch?time_continue=3&v=th7jZlmoZBc

Developers

- ▶ N NEO C# Developers Center of Excellence (neo-csharpcoe)
<https://github.com/mwherman2000/neo-csharpcoe/blob/master/>
- ▶ Discord NEO Community (like Slack)
<https://discord.gg/gqCYeup>
- ▶ The NEO Project
<https://github.com/neo-project>
- ▶ City of Zion awesome-neo project: Developer Docs
<https://github.com/CityOfZion/awesome-neo/#developer-documentation>



NEO Blockchain C# Developers Center of Excellence

<https://github.com/mwherman2000/neo-csharpcoe>

- ▶ Founder: Michael Herman (neotoronto@outlook.com)
- ▶ NEO Blockchain end-to-end, detailed guidance for .NET developers
- ▶ Content purposely designed to help .NET developers ramp up and become productive more quickly
- ▶ Repository for content presented at the NEO C# NET Developers Center of Excellence meetup group
- ▶ Contents
 - ▶ NEO Blockchain Quick Start Guide for .NET Developers
<https://github.com/mwherman2000/neo-dotnetquickstart>
 - ▶ NEO Blockchain Architecture Reference Model (ARM) for .NET Developers (neo-charm)
<https://github.com/mwherman2000/neo-charm>
 - ▶ NEO C# NET Developers Center of Excellence
<https://github.com/mwherman2000/neo-windocs/tree/master/events/2018-neo-blockchain-toronto>

NEO Blockchain Quick Start Guide for .NET Developers

<https://github.com/mwherman2000/neo-dotnetquickstart/blob/master/README.md>

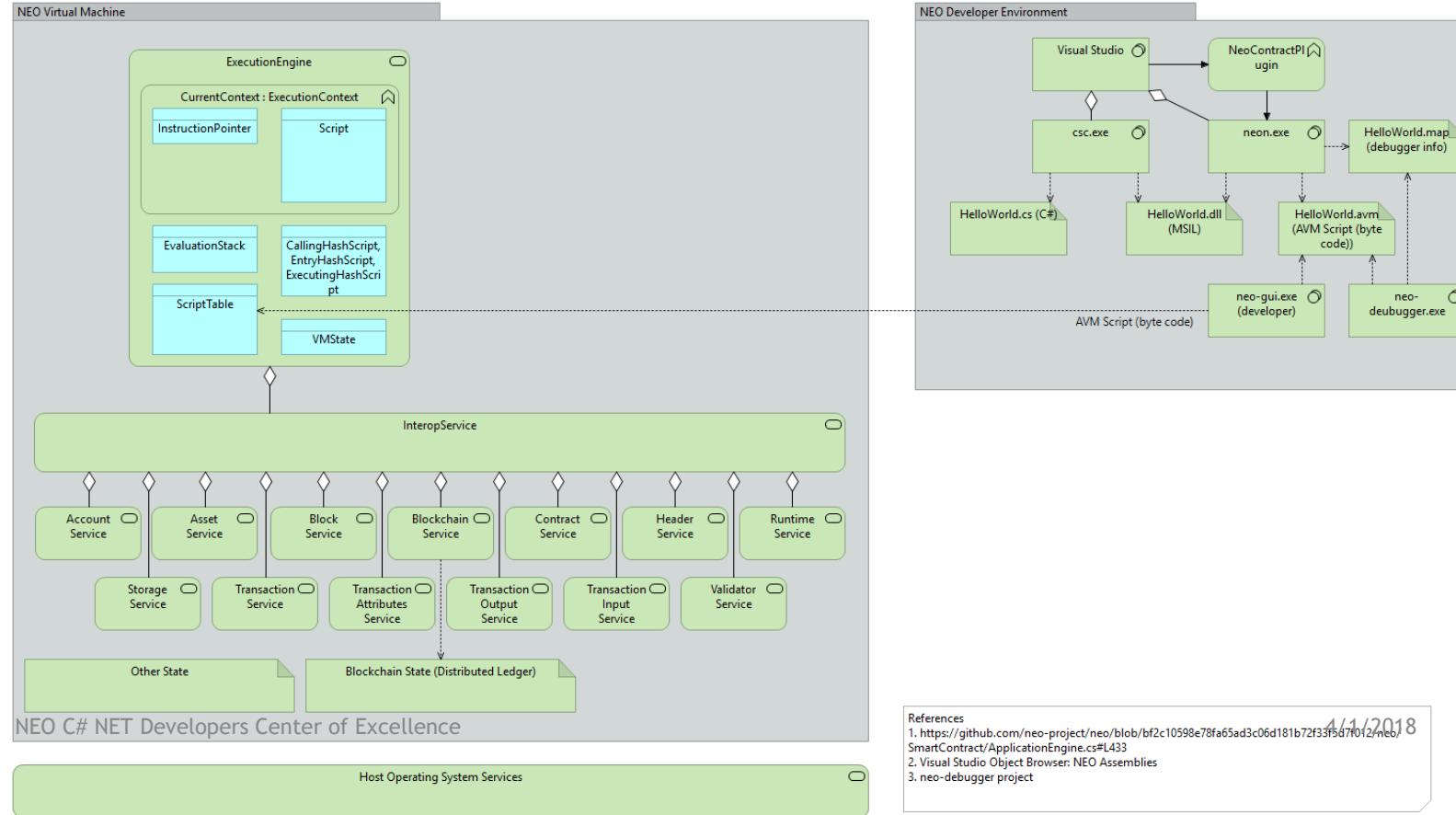
1. Pre-requisites and Recommendations
 2. Download and install Visual Studio 2017 Community Edition integrated development environment (IDE)
 3. Download and unpack NEO developer tool projects (source)
 4. Coffee time: Wait for previous activities to complete
 5. Install NeoContractPlugin Visual Studio extension
 6. Build and test NEO developer tool projects (from source)
 7. Download, install, and test Docker platform
 8. Download and test NEO privatenet Docker container
 9. Create and compile HelloWorld smart contract sample
 10. Deploy and test the HelloWorld smart contract
 11. Celebrate
- ▶ Appendix A - Checklist
 - ▶ Appendix B - Roadmap
 - ▶ Appendix C - Reset NEO privatenet Environment: Container, Wallets, and Clients

Statistics

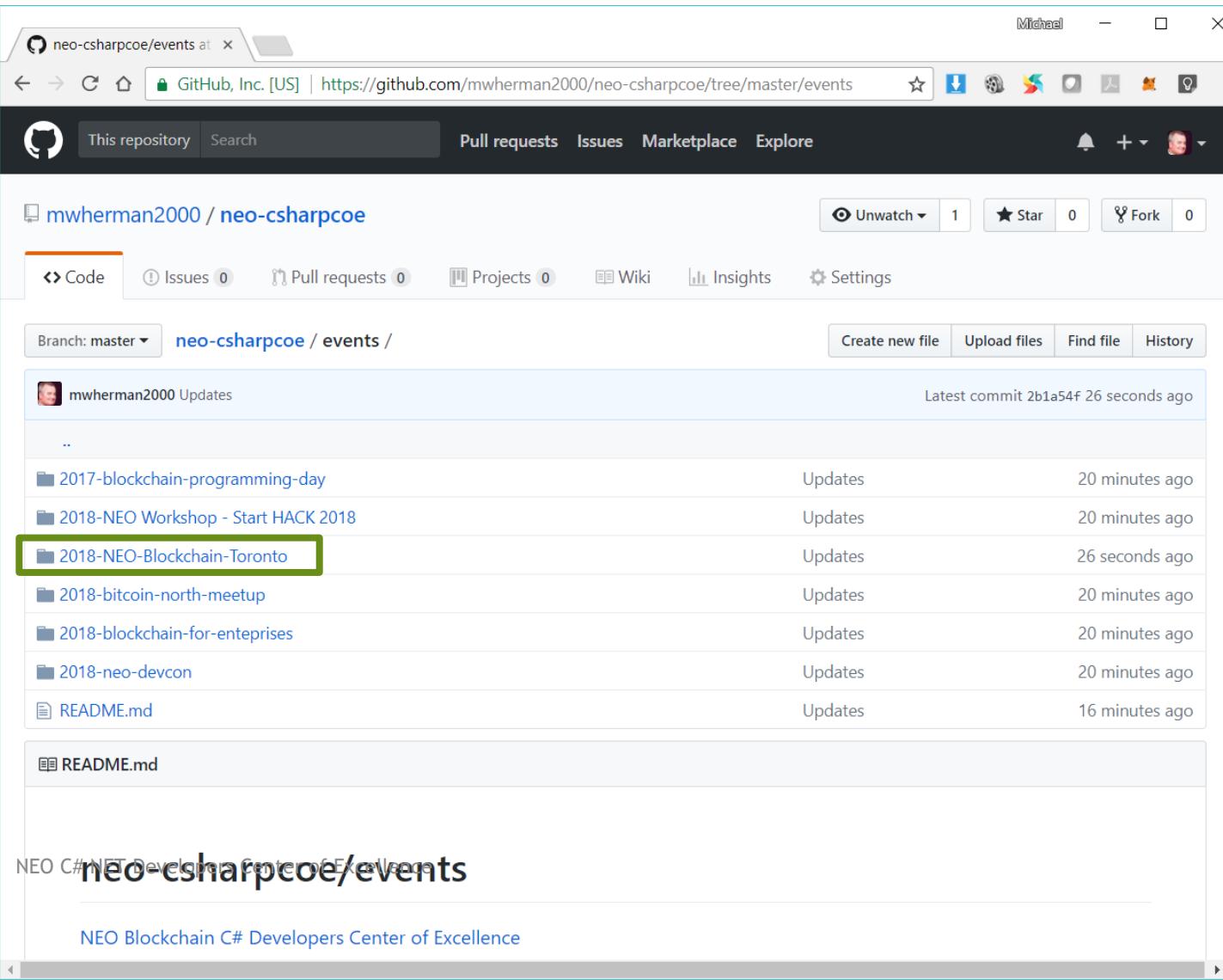
- ▶ 10 activities comprising 130 documented tasks (approximately)
- ▶ 140 screen shots (approximately)
- ▶ 7 batch files
- ▶ 2 JSON configuration files
- ▶ 1 C# code snippet

NEO Blockchain Architecture Reference Model (ARM) for .NET Developers (neo-charm)

<https://github.com/mwherman2000/neo-charm/blob/master/README.md>



NEO C# .NET Developers Center of Excellence / NEO Blockchain Toronto



neo-csharpcoe/events at Michael X

[GitHub, Inc. \[US\]](#) | <https://github.com/mwherman2000/neo-csharpcoe/tree/master/events>

This repository Search Pull requests Issues Marketplace Explore

mwherman2000 / neo-csharpcoe Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master neo-csharpcoe / events / Create new file Upload files Find file History

File/Folder	Type	Last Commit
mwherman2000 Updates	Updates	Latest commit 2b1a54f 26 seconds ago
..		
2017-blockchain-programming-day	Updates	20 minutes ago
2018-NEO Workshop - Start HACK 2018	Updates	20 minutes ago
2018-NEO-Blockchain-Toronto	Updates	26 seconds ago
2018-bitcoin-north-meetup	Updates	20 minutes ago
2018-blockchain-for-enterprises	Updates	20 minutes ago
2018-neo-devcon	Updates	20 minutes ago
README.md	Updates	16 minutes ago

README.md

neo-csharpcoe/events

NEO C# .NET Developers Center of Excellence

NEO Blockchain C# Developers Center of Excellence

be/tree/master/events

4/1/2018

99



Global Blockchain Compliance Hub

[Legal Disclaimer](#)

[ABOUT](#)

[BELGIUM](#)

[GERMANY](#)

[GIBRALTAR](#)

[HONG KONG](#)

[JAPAN](#)

[LUXEMBOURG](#)

[NETHERLANDS](#)

[SINGAPORE](#)

[SOUTH KOREA](#)

[SWITZERLAND](#)
NEO C# .NET Developers Center of Excellence

[THE ISLE OF MAN](#)

Welcome!

You've come to NEO's Global Blockchain Compliance Hub, an ongoing project aimed at compiling a public Github database that includes legal details related to blockchain and cryptocurrency for as many jurisdictions as possible worldwide. We're hoping this can become a community-driven community resource that will help folks interested in starting their own NEO-based blockchain projects think about regulation as they move towards the point where they'd need to actually seek counsel. We're undertaking this project because NEO is committed to compliant and sustainable growth, and we believe this sort of resouce is currently lacking.

If you're interested in contributing (by either updating a country page or creating a new one), please let us know - we're always looking for new partners. You can reach out to milesgraham@neo.org.

For each jurisdiction, we aim to answer the following information:

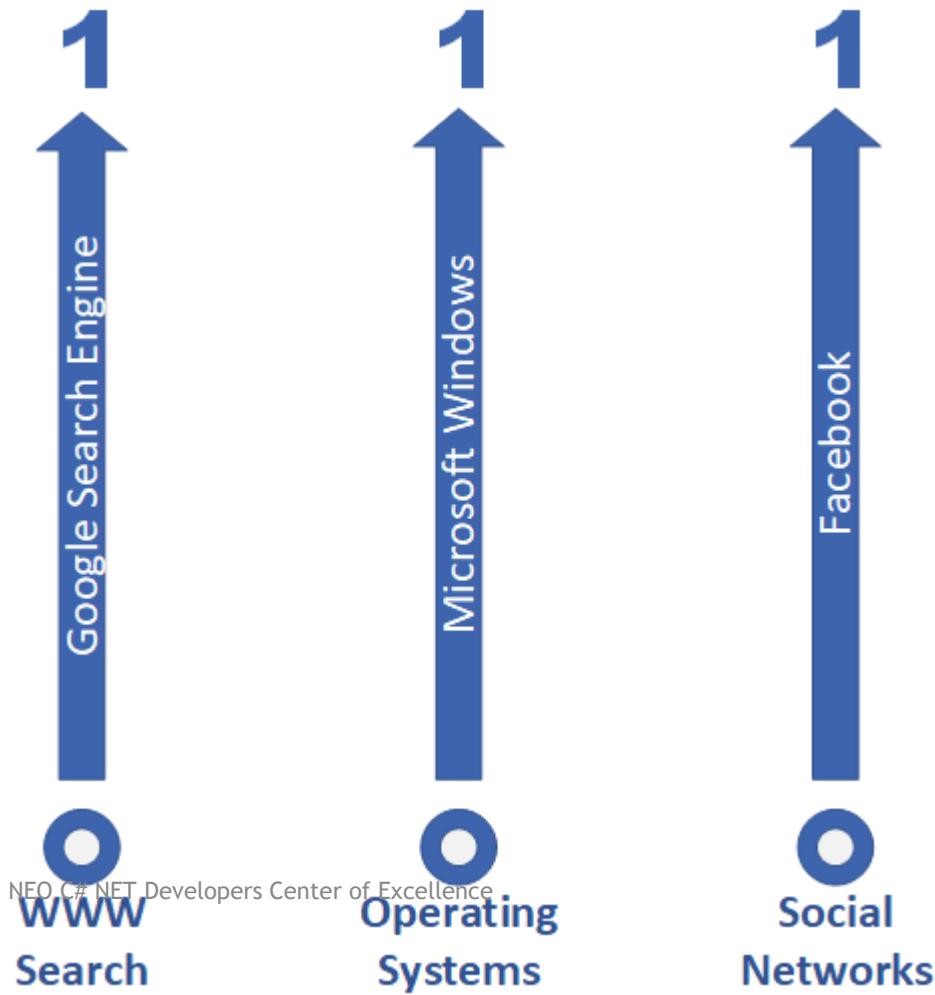
1. Non-profit/for-profit company registry requirements
2. Team member nationality requirements
3. Tax & auditing requirements
4. Governing laws/bylaw requirements
5. Laws related to token sales, blockchain, and digital proof
6. Securities-related laws
7. Privacy & data protection-related laws
8. Who bears final responsibility/liability in case something goes wrong?
9. What is a smart contract and is it legally binding?
10. How is dispute resolution integrated?
11. Can smart contracts be nullified?
12. Suggested readings

Why is the NEO Blockchain important?

“The NEO Blockchain will as important or more important than Windows for enterprise distributed application development.”

[Michael Herman, March 2018]

“Zero to One” by Peter Thiel, Paypal Founder



4/1/2018

102

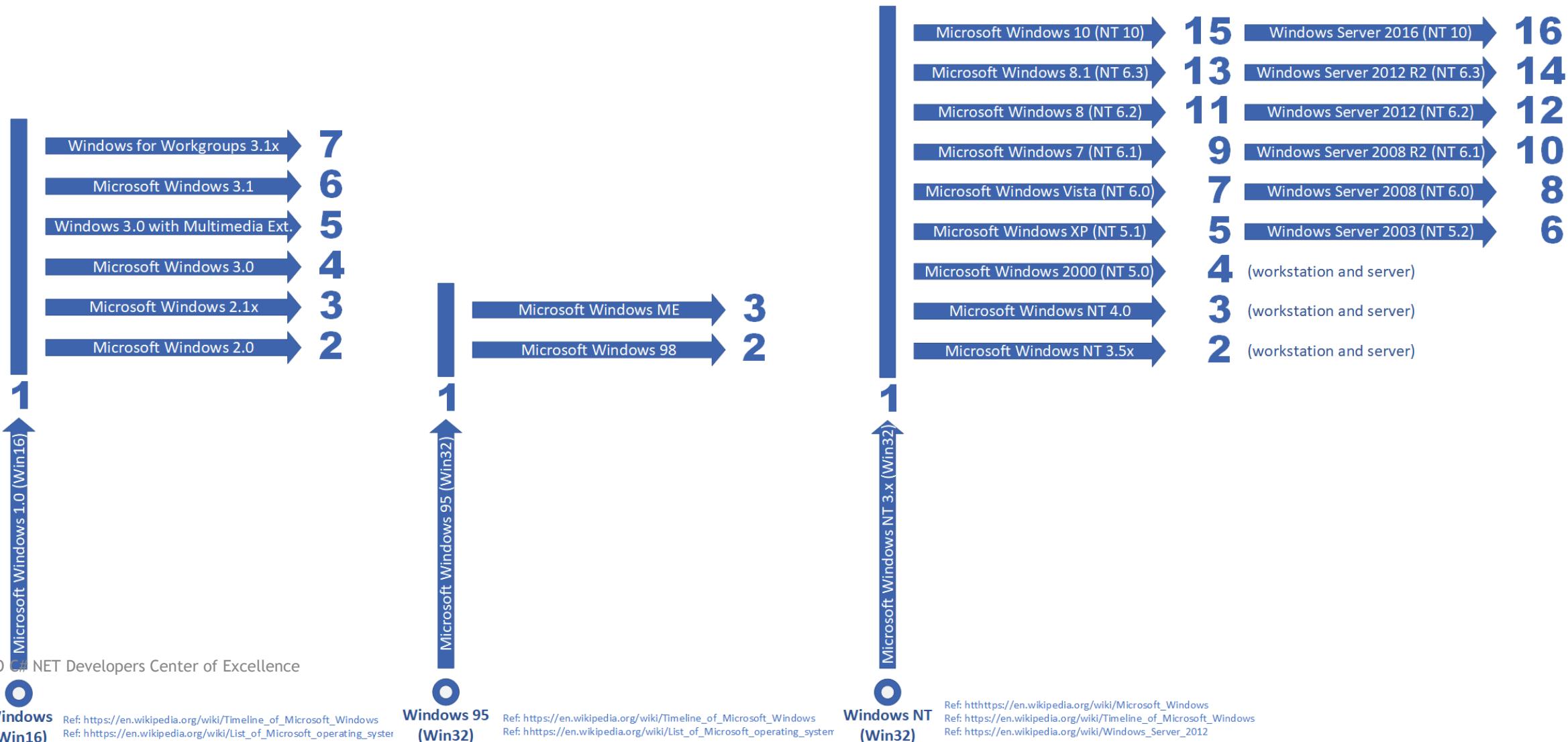
Zero to One: Microsoft Windows

Microsoft Windows (Win16) ↑ 1

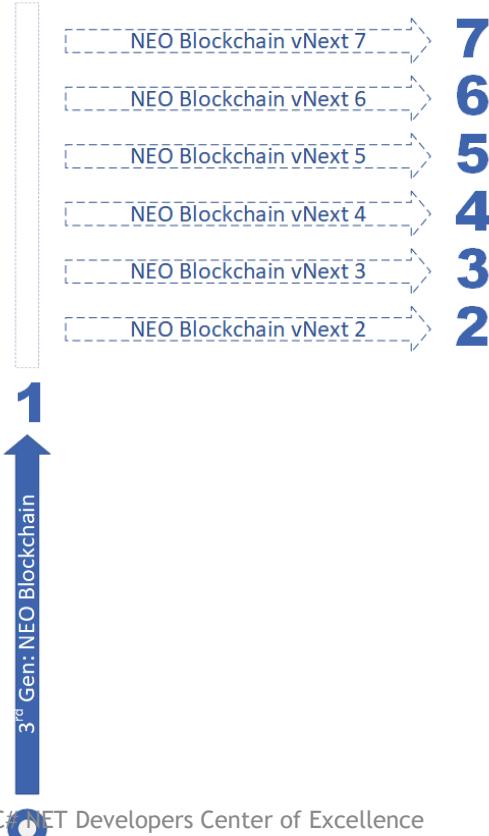
Microsoft Windows NT 3.x (Win32) ↑ 1

Microsoft Windows 95 (Win32) ↑ 1

Zero to One: Microsoft Windows



Zero to One: NEO Blockchain Platform



NEO C# .NET Developers Center of Excellence

Blockchain
Technology
(Enterprise dApps)

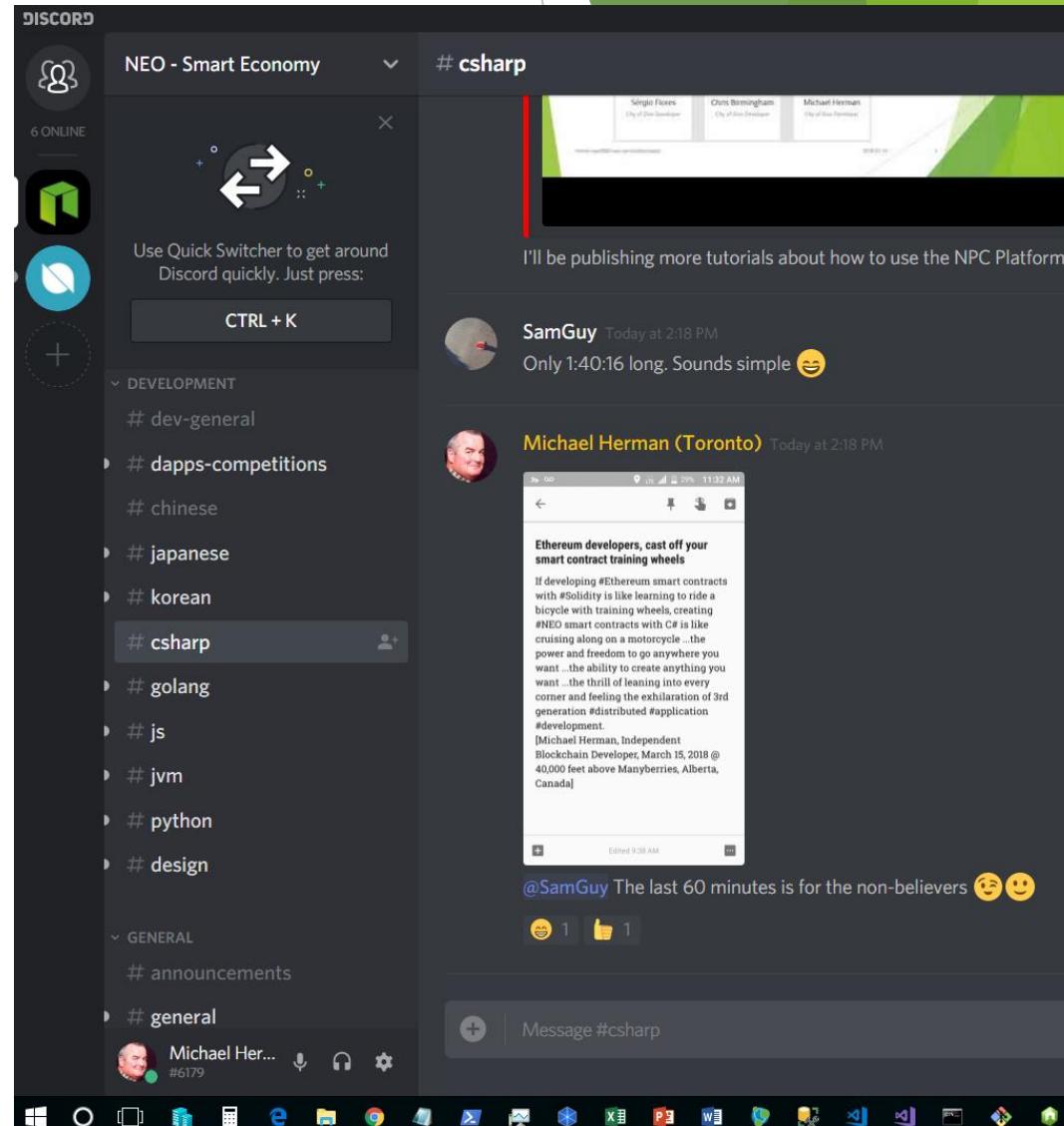
4/1/2018

105

8. Next Steps

Next Steps

1. Join Discord
<https://discord.gg/gqCYeup> /
<https://discord.gg/4TQujHj>
2. Read through and contribute to NEO C# Developers Center of Excellence (neo-csharpcoe)
<https://github.com/mwherman2000/neo-csharpcoe/blob/master/README.md>
3. Start with NEO Blockchain Quick Start Guide for .NET Developers
<https://github.com/mwherman2000/neo-dotnetquickstart/blob/master/README.md>



NEO BLOCKCHAIN ENTERPRISE DEVELOPER (CNED) CERTIFICATION PROGRAM

PROGRAM OVERVIEW

This program will guide the participants through the basics of the NEO Blockchain and the principles of the NEO Smart Economy, while mastering the best practices for NEO Blockchain enterprise application development.

By completing the program, participants will be able to create and test their own distributed applications (dApps); while also learning about the best tools and reusable code libraries, frameworks, how-to examples, and best practices for enterprise application development using .NET/C#, C#.NEO, and the NEO Blockchain platform.

PROGRAM DETAILS

Program Length

3 days
plus capstone project

Date and Time

TODO-TODO

9:00 a.m. - 5:00 a.m.

Location

4/1/2018

108

Bergeron Centre for Engineering
Excellence, York University

Reminder: When to use Blockchain Technology

“Blockchain: the force multiplier for the smart economy” [Microsoft]

Questions to Ask

1. Is it a business process that crosses trust boundaries?
2. Do multiple parties manipulate the same data?
3. Are processes operating inefficiently or decisions delayed due to the number of intermediaries?
4. Does the business process involve low-value, manual verification steps?

Use a Blockchain when there is a Need for

- ▶ Real-time transparency by connecting business processes across organizations
- ▶ Real-time, transparent access to a verifiable source of the truth across organization boundaries
- ▶ Introduce trust and increase efficiency amongst participants - reducing the need for intermediaries
- ▶ Improved efficiencies and increased confidence through automation and smart contracts that execute consistently.

“Use blockchain technology when it matches the requirements of your application. Don’t automatically use blockchain technology simply because you’re re-platforming an existing application.” [Gartner 2018]

Tokenize Every Little Thing (ELT)

Just over one year ago, I introduced the concept of *graphitization* and talked about [#Graphitization of the Enterprise](#). I opened the article with the challenge:

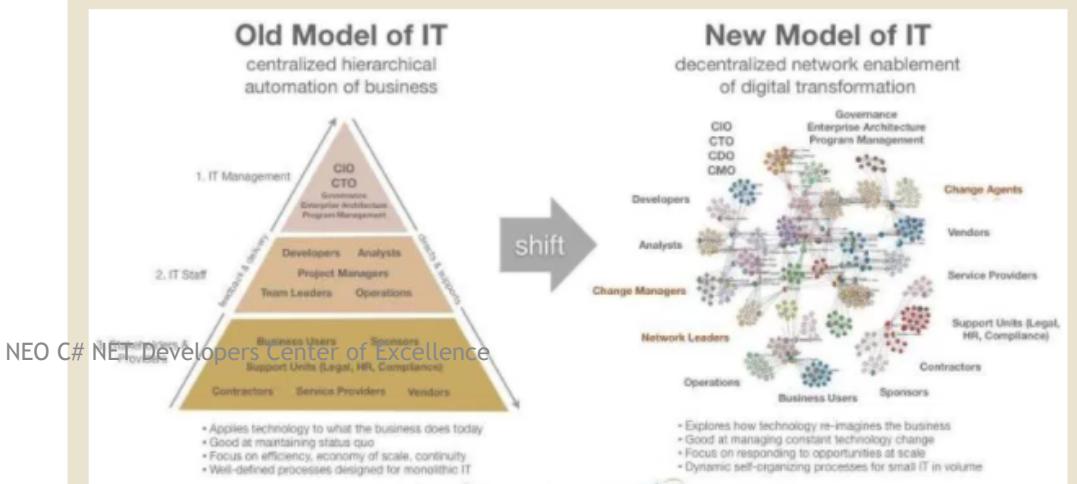
Move beyond digitalization of the enterprise to graphitization of the enterprise.

For 2018 and beyond, the challenge is simpler but more difficult:

Tokenize Every Little Thing (ELT)

To provide more context, let me first quote from the introductory paragraphs of the [#Graphitization](#) article.

Here's a great diagram that explains this concept [graphitization]. (click on the [diagram](#) to enlarge it)



- Tokenize Every Little Thing (ELT)
- Ethereum geth.exe Dashboard built with Nethereum
- Ethereum .NET Development Examples using the Nethereum Integration Libraries
- Michael Herman, Blockchain Developer, Enterprise Architect and Data Scientist:
#Graphitization Inventor
- #Graphitization of the Amazon Leadership Principles (introducing Personal Leadership Principle Maps)
– Iteration 1
- Why Would You Prefer to Work for Amazon (or Facebook) over Microsoft (or Salesforce)? [WIP]
- High-Velocity Service Packages and Envelopes [WIP]
- Isomorphic Weighted Graph Databases and Graph Algorithm Non-Collinearity [WIP]
- What are the differences between improving the design (and operation) of an aircraft engine, a muscle car, a large enterprise, and/or an integrated commercial global cloud services platform ...all running at hyperscale?
- Structuring Small Powerful Documents

Categories

- .NET Development

I leave you with a challenge...

Become part of the NEO Smart Economy

**TOKENIZE
EVERY LITTLE THING
#ELT**



hyperonomy.com - digital intelligence (my blog)

<https://hyperonomy.com/2018/01/24/tokenization-of-every-little-thing-elt/>

Questions?

Michael Herman (Toronto) - Independent Blockchain Architect and Developer

E: <mailto:neotoronto@outlook.com>

M: <https://www.meetup.com/NEO-Blockchain-Toronto>

G: <https://github.com/mwherman2000/neo-csharpcoe/tree/master/events>

F: <https://www.facebook.com/neotoronto/>

T: <https://www.twitter.com/neotoronto>

Backup Slides

Our Values as *Neoxplorers*

We operate like the NEO Blockchain

- ▶ Decentralized
- ▶ Community based
- ▶ Consensus driven
- ▶ Creators of value
- ▶ Store of value (knowledge)
- ▶ Transfer of value (training)
- ▶ Multi-disciplinary
- ▶ Multi-application
- ▶ Multi-technology

We behave like the NEO Blockchain

- ▶ Robust
- ▶ Reliable
- ▶ Transparent
- ▶ Smart
- ▶ Open
- ▶ Non-discriminatory
- ▶ Participatory
- ▶ Diversity
- ▶ Committed

NEO Persistable Classes (NPC) YouTube Channel

https://www.youtube.com/playlist?list=PLU-rWqHm5p46bIDXPNf4c2JP_AOkpnV5

The screenshot shows the YouTube channel page for "NEO Persistable Classes (NPC)". The channel has 7 videos and was updated today. The channel description is: "A Very Efficient Object-Oriented dApp Platform for Global Scale Smart Economy Applications". The channel art is a green and white graphic for the "neo-csharpco Community Blueprint". The channel owner is Michael Herman.

Channel Statistics:

- 7 videos
- Updated today

Channel Art:

Channel Description:

A Very Efficient Object-Oriented dApp Platform for Global Scale Smart Economy Applications

Recent Videos:

- How to run NPCdApp 1.0 - CoZ dApp Competition #2 [update 4] (Michael Herman, 13:46)
11. Quick Cycle Edit-Compile-Debugging of C#.NEO Smart Contracts (Michael Herman, 12:29)
13. NEO Persistable Classes (NPC) NEP5 Token Template: Preview (Part 1) (Michael Herman, 37:53)
12. NEO Persistable Classes (NPC) Platform 2.1: Preview (Michael Herman, 35:55)
- NEO Persistable Classes (NPC) Platform 2.0: Deep Dive (Michael Herman, 1:40:17)
- NEO Persistable Classes 1.0: Deep Dive (Video 2 of 3) [Update 1] (Michael Herman, 1:24:17)
11. Quick Cycle Edit Compile Debugging: Single Stepping and Profiling (Michael Herman, 26:42)

Our Values as *Neoxplorers*

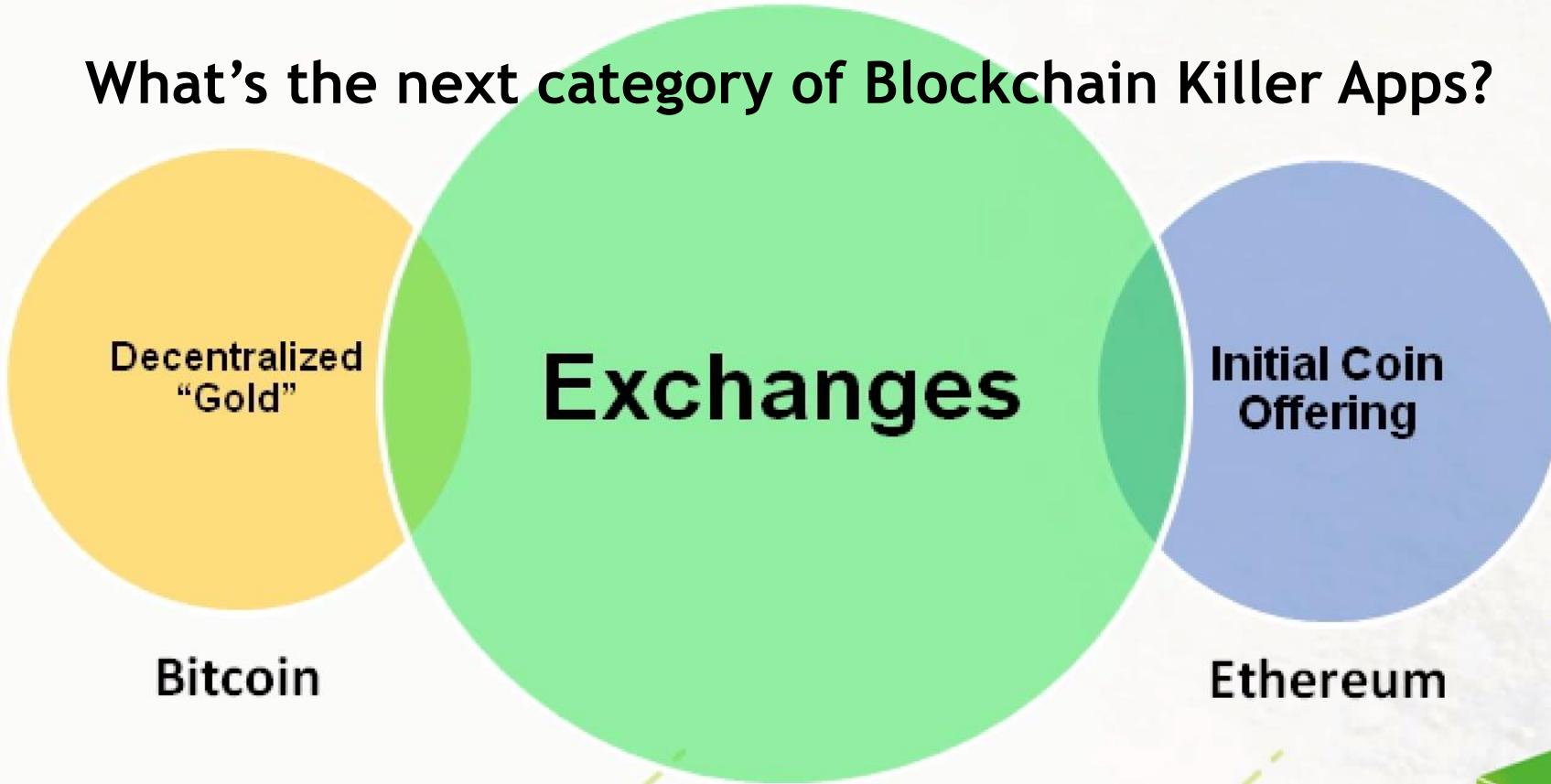
We operate like the NEO Blockchain

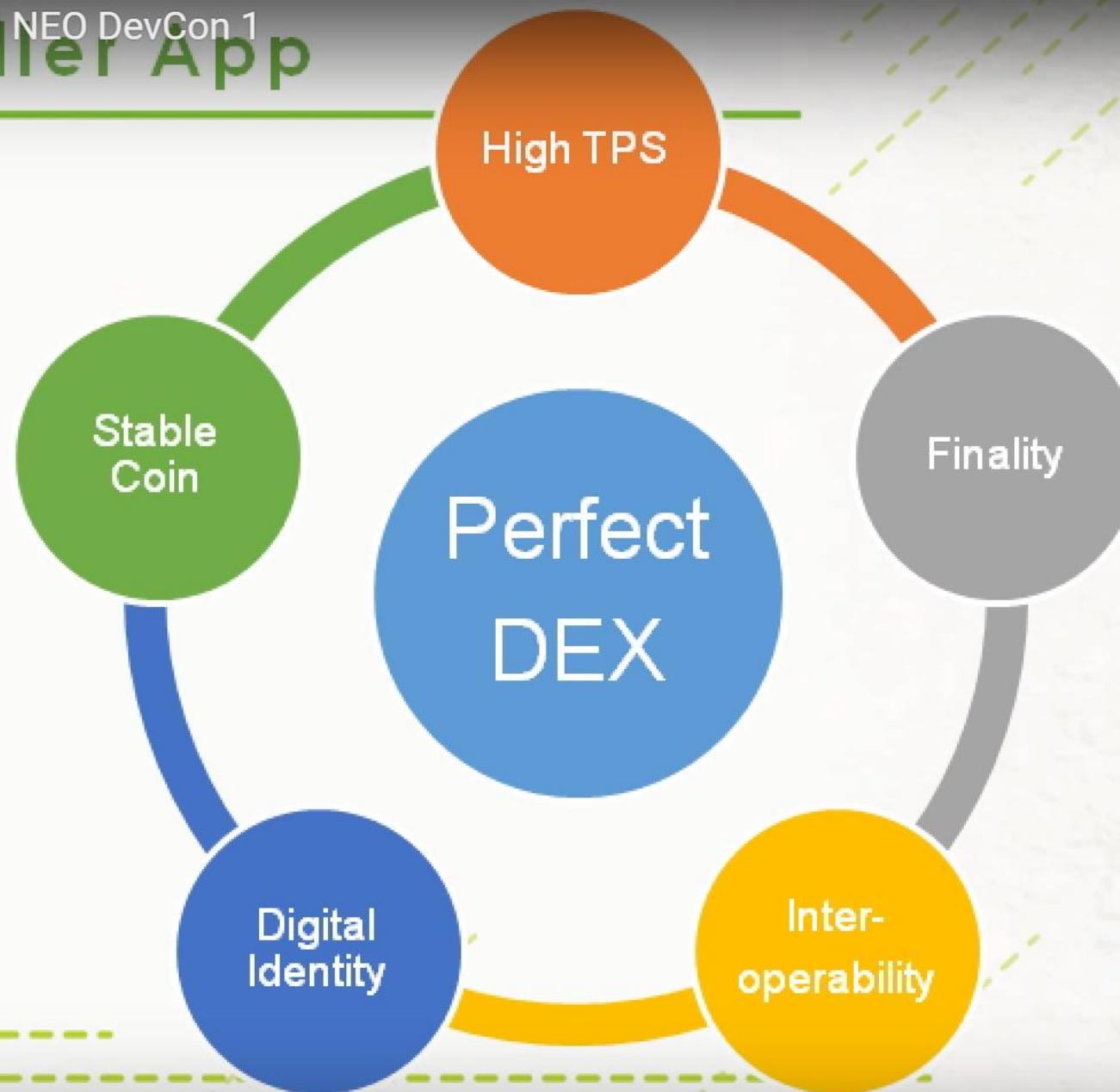
- ▶ Decentralized
- ▶ Community based
- ▶ Consensus driven
- ▶ Creators of value
- ▶ Store of value (knowledge)
- ▶ Transfer of value (training)
- ▶ Multi-disciplinary
- ▶ Multi-application
- ▶ Multi-technology

We behave like the NEO Blockchain

- ▶ Robust
- ▶ Reliable
- ▶ Transparent
- ▶ Smart
- ▶ Open
- ▶ Non-discriminatory
- ▶ Participatory
- ▶ Diversity
- ▶ Committed

What's the next category of Blockchain Killer Apps?





NEO C# .NET Developers Center of Excellence



WHERE SHOULD
WE FOCUS
THIS YEAR?

"BLOCKCHAIN"



IT WILL
CHANGE
EVERYTHING.



EVERYBODY
IS TALKING
ABOUT IT.



THE POTENTIAL
APPLICATIONS
ARE ENDLESS.

WE DON'T
WANT TO BE
LEFT BEHIND.

WHAT
EXACTLY IS
BLOCKCHAIN?

ALSO,
"ARTIFICIAL
INTELLIGENCE"

