NEO Community Bluepaper

NEO Persistable Classes 1.0 Deep Dive

An Efficient Object-Oriented Framework for C#.NEO Smart Contract Development

Michael Herman

https://github.com/mwherman2000/neo-persistibleclasses

neotoronto@outlook.com

NEO smart economy

What is a Bluepaper?

- From: Blue Papers – Writing Technical Specifications
  http://www.genuinewriting.com/blue_paper.html

  *The term "blue paper" is a currently seldom used phrase that is slowly catching on around the world. First used in Germany, the term is used as a means of distinguishing between policy and procedure papers (white papers) and those **papers outlining only technical specifications and descriptions of new technologies or particular pieces of equipment**.*

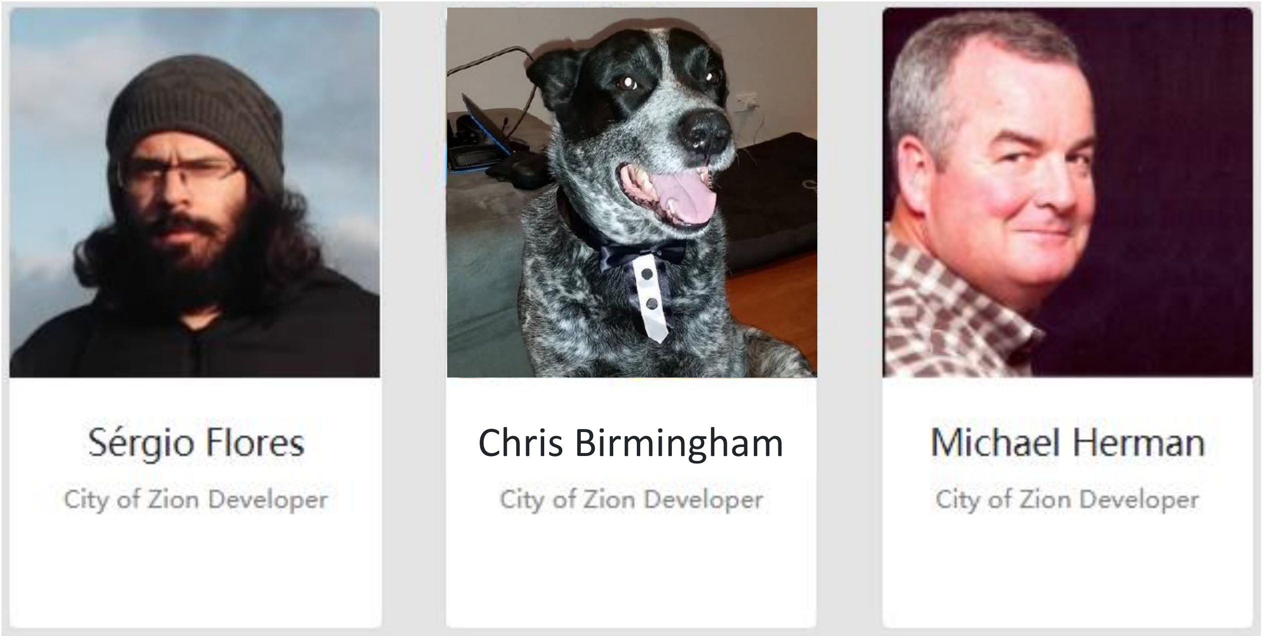
- What is a NEO Community Bluepaper?

  It is a bluepaper written by anyone who is passionate about the NEO Blockchain platform. They can be unsponsored; or sponsored by someone else or an organization (e.g. an employer).

- The NEO Persistable Classes 1.0 Deep Dive is an unsponsored bluepaper.

Shout-outs

- NEO Developer Tool Suite - CoZ Developers
  - @relfos – neo-debugger
  - @birmas – neo-gui-developer
  - @mwherman2000 – event log enhancements



Agenda

1. Purpose
2. Scenario
3. Problem
4. Analysis
5. Options
6. Code Solution
7. Test Results
8. Roadmap
9. Summary
10. Appendix A - NEO Blockchain Architecture Reference Model (ARM)

Checkout *Structuring Small Powerful Documents* at https://hyperonomy.com/2017/04/09/structuring-small-powerful-documents/

0. Key Definitions

- NPC
  - NEO Persistable Class Framework
- NeoVersionedAppUser (VAU)
  - NEO Versioned App User objects used to create NeoStorageKeys
- NeoStorageKeys (NSK)
  - NEO Storage Keys used to implement NPC Level 4 Collectible NPC classes
- NEO-KONG
  - NEO Key-Object Notation for Geeks is a specification used to implement NSKs
- C#.MS
  - Version of C# supported natively by the C# compiler in Visual Studio 2017 (C# 6.0)
- C#.NEO
  - The label for the implicit (undocumented) definition what is supported by the transcompiler and the NEO VM
  - Subset of C#.MS supported by the NEO neon.exe transcompiler (as of January 2018)
- C#.NPC
  - Explicit definition of the subset of C#.MS that is supported by the NEO transcompiler for .NET (neon.exe) that supports NPC's best practice recommendations for object-oriented development.

1. Purpose

- The purpose of this presentation is to explain the key drivers for the NEO Persistable Class Framework 1.0 (NPC) as well as a range of design and implementation details including:
  - A fairly complete description of C#.NEO, the subset of C#.MS (C# 6.0) supported by neon.exe, the C#.MS MSIL to NEO byte code transcompiler
  - Strategies and options for supporting the object programming capabilities found in C#.MS but not supported in C#.NEO
  - Basic guidance to smart contract development using C#.NPC
  - NPCdApp, the initial reference implementation for the NEO Persistable Class Framework

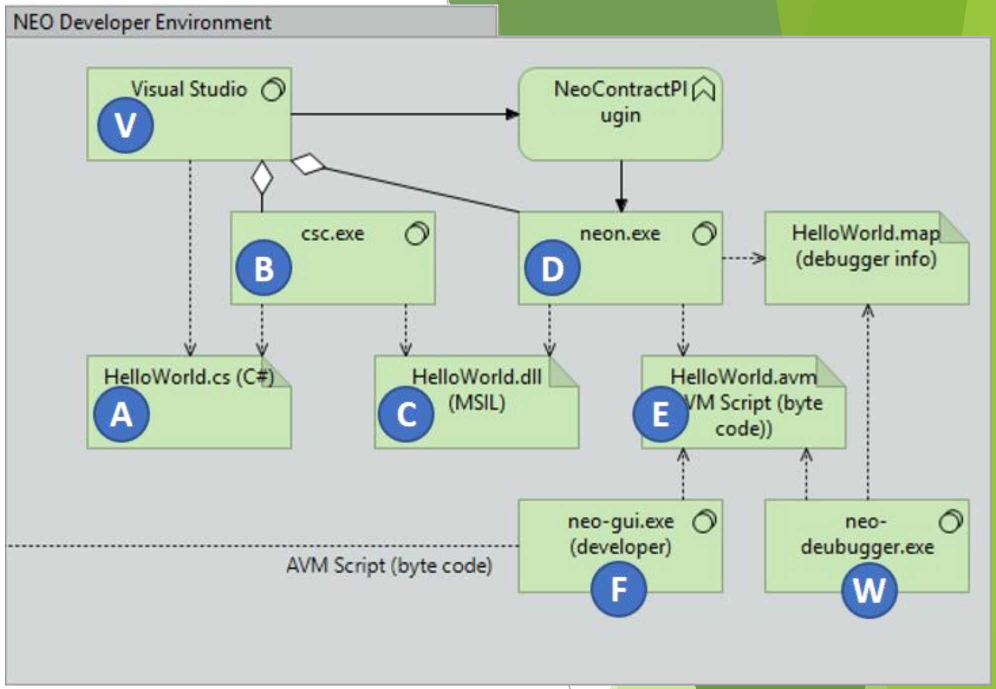
- Inspiration (NPC Level 0)
  - StructExample example https://github.com/neo-project/examples-csharp/blob/master/StructExample/StructExample.cs

2. Scenario

2. Scenario

- Developers on the NEO Blockchain platform have a variety of languages, tools, privatenet and testnet environments as well other technology options to consider for developing smart contracts
- Language options include:
  - C#.NEO
  - Python
  - Java
  - and others
- As a scenario, this document specifically targets C#.NEO smart contract development using Visual Studio 2017

2. Scenario (con't)



The diagram depicts a typical NEO development environment.

- A Developers develop their smart contract's code using their favorite IDE and high-level programming language (HLL) (e.g. C#.MS)
- C The IDE's integrated compiler compiles the HLL source into intermediate code (IL) (e.g. Microsoft Intermediate Language)
- E A transcompiler then translates the IL into byte code that is executable by the NEO VM (e.g. NEO VM byte code script files)
- W A debugger may be used to execute the VM byte code script off-chain using a VM emulator
- F An on-chain deployment and testing tool may be used to further test and validate the smart contract's implementation

- Reference: NEO Blockchain Architecture Reference Model (ARM) for .NET Developers https://github.com/mwherman2000/neo-charm

2. Scenario (con’t)

- The NEO Development environment gains incredible value from leveraging the Microsoft .NET compiler stack but there is a significant challenge with this approach:
  - Visual Studio doesn't know anything the instruction set that the NEO VM supports
  - The C#.MS compiler implements the full C# 6.0 specification and knows how to generate MSIL (MS Intermediate Language) code (DLL assembly files) for the full C#.MS language
  - On the other hand, the C#.NEO compiler (neon.exe) only supports a subset of the MSIL patterns generated by the MS compiler.
  - neon.exe reads MSIL files and transcompiles the logic into NEO byte code script files (.AVM files) and generates an error when an unsupported MSIL pattern is encountered
- I call this subset of C#.MS that is ultimately supported in the NEO VM: C#.NEO
  - What are the supported patterns of MSIL that neon.exe supports?
  - What are the good/better/best patterns to use?
  - How do you work around what is important but not supported in C#.NEO?

3. Problem

3. Problem

- Given that the subset of C#.MS that is supported by neon.exe (and the NEO VM) is not documented, how to we proceed to exploit the capabilities of C#.NEO?
  - What are the supported patterns of MSIL that neon.exe supports?
  - What are the good/better/best patterns to use?
  - How do you work around what is important but not supported in C#.NEO?
- How do we use C#.NEO for sophisticated object-oriented development?
- How do we support C# classes that are nothing more elaborate than C++/C## structs?

```
public class Point /* Level 0 */
{
    private BigInteger _x;
    private BigInteger _y;
}
```

4. Analysis

Sidebar: How is programming the NEO VM like learning to fly a flight simulator?

- As an analogy, learning to program the NEO VM is like learning to fly a flight simulator.
- A NEO smart contract running in the NEO VM is like the pilot in control of the flight simulator.
- The smart contract is performing various calculations and calling various virtual machine APIs to accomplish the smart contract's purpose and goals.
- In response to these API calls, the VM is interacting with its various subsystems (just like the flight simulator).
- https://github.com/mwherman2000/neo-charm/blob/master/README.md

4. Analysis

- What works in C#.NEO? What doesn't? What are the (good) workarounds?

C#.MS	C#.NEO	C#.NPC
Class constructor (parameterless)	Supported: field layout only	`public static Point New()` class method `private static _Initialize(p)` class method
Class constructor (with parameters)	Not supported: ignored	`public static Point New(x,y)` class method
Class constructor logic	Not supported: ignored	`public static bool Foobar(p,a,b)` class methods
Properties (`get()/set()`)	Not supported	`public static SetX/GetX/Set()` class methods
`null` valued variables	Not supported	`EntityState _state; //` system field `public static bool IsNull(p)`
Floating-point data types	Not supported	Native Storage datatypes: `string, byte[], BigInteger` Limited datatypes: `int/Int32`
Integer data types (`int/Int32`)	Partially support	
`switch` statements	Not predictable	Use `if-then-else if-then-else if-…`

4. Analysis (con't)

C#.MS	C#.NEO	C#.NPC
Declare fixed size `byte[100]`	Not supported	Use byte[].Concat(byte[])
Assignment to element of `byte[]` array: `a[i] = 0x12;`	Not supported	Use byte[].Concat(byte[])
`0x0123456789abcdef.ToString()`	Not supported	`Neo.SmartContract.Framework.Helper.AsString(byte[])`
`((BigInteger)0x0123456789abcdef).ToString()`	Not supported	
`((BigInteger)0x0123456789abcdef).ToByteArray().ToString()`	Not supported	`((BigInteger)0x0123456789abcdef).AsByteArray().AsString()`
`enum` enumerations at `int`'s (internally)	Supported	`int/Int32` are supported but not native C#.NEO Storage datatypes NOTE: Be prepared to do lots of type coercions

4. Analysis (con't)

```
namespace Neo.SmartContract.Framework
{
    public static class Helper
    {
        ...public static BigInteger AsBigInteger(this byte[] source);
        ...public static byte[] AsByteArray(this BigInteger source);
        ...public static byte[] AsByteArray(this string source);
        ...public static string AsString(this byte[] source);
        ...public static byte[] Concat(this byte[] first, byte[] second);
        ...public static byte[] HexToBytes(this string hex);
        ...public static byte[] Range(this byte[] source, int index, int count);
        ...public static byte[] Take(this byte[] source, int count);
        ...public static Delegate ToDelegate(this byte[] source);
        ...public static byte[] ToScriptHash(this string address);
    }
}
```

4. Analysis (con’t)

```
namespace Neo.SmartContract.Framework.Services.Neo
{
    public static class Storage
    {
        public static StorageContext CurrentContext { get; }

        ...public static void Delete(StorageContext context, byte[] key);
        ...public static void Delete(StorageContext context, string key);
        ...public static byte[] Get(StorageContext context, byte[] key);
        ...public static byte[] Get(StorageContext context, string key);
        ...public static void Put(StorageContext context, byte[] key, byte[] value);
        ...public static void Put(StorageContext context, byte[] key, BigInteger value);
        ...public static void Put(StorageContext context, byte[] key, string value);
        ...public static void Put(StorageContext context, string key, byte[] value);
        ...public static void Put(StorageContext context, string key, BigInteger value);
        ...public static void Put(StorageContext context, string key, string value);
    }
}
```

4. Analysis (con’t)

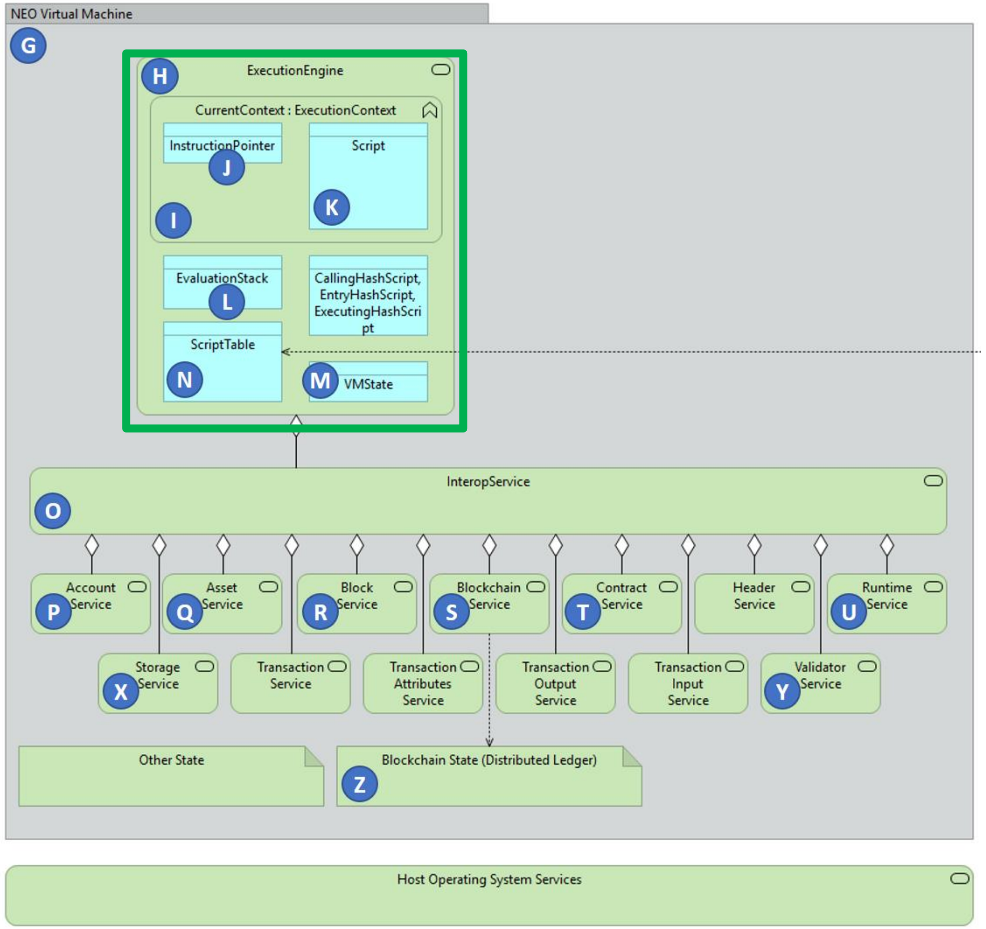
NEO Account Addresses, Keys and Key Lengths

Description	Length (byte[]/HexString)	Examples (Bold Lengths)
WIF1	(**52**/104)	L3f7C21q4Mu5FzZsDuCMeHqwJ1apHYCrwzU2821p1opaM43BAMKo
WIF1Address	(**34**/68)	AcCHoikUq9cP6SMESHufCEMwADJNcTwnAv
WIF1PublicKeyHex	(33/**66**)	02c44534465c8b21f659eba5708e69edae1ddd6f8cd63004095f8e39493cf54e82
WIF1PrivateKeyHex	(32/**64**)	c016e1c8a193cc1a28a15464106b91b52727547a3a36f40a8bfebd9933d1963c
WIF1AddressScriptHash	(20/**40**)	e000aa6a0ab08af8aa78b19d481e5b5c40d8be0e
	= WIF1Address.AsScriptHash();	
WIF2	(**52**/104)	KxDgvEKzgSBPPfuVfw67oPQBSjidEiqTHURKSDL1R7yGaGYAeYnr
WIF2Address	(**34**/68)	AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y
WIF2PublicKeyHex	(33/**66**)	031a6c6fbbdf02ca351745fa86b9ba5a9452d785ac4f7fc2b7548ca2a46c4fcf4a
WIF2PrivateKeyHex	(32/**64**)	1dd37fba80fec4e6a6f13fd708d8dcb3b29def768017052f6c930fa1c5d90bbb
WIF2AddressScriptHash	(20/**40**)	23ba2703c53263e8d6e522dc32203339dcd8eee9
	= WIF2Address.AsScriptHash();	

TxID, AssetID	(32/**64**)	687b68a1159429dc558e4fc7590e391d52f1ef79a12922f941daa37c00334ec5

Sidebar: NEO Virtual Machine

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



- This Analysis section has dug relatively deep in terms of analysing the core ExecutionEngine in the NEO VM from a C# development perspective.
- It has started to create a definition of C#.NPC as an explicit definition of the subset of C#.MS that is supported by the NEO transcompiler for .NET (neon.exe) **and** follows recommended and documented best practices.
- (C#.NEO remains as the label for the implicit (undocumented) definition what is supported by the transcompiler and the NEO VM).

5. Options

5. Options

Model every C# Class as an C#.NPC Class

- Fields
  - Only `string, byte[],` or `BigInteger` data types
  - Especially if there is any chance of class being persisted
- Constructors
  - Every class has a private parameterless constructor that is never used
  - Every class has `public static T New()` and `public static T New(x,y)`
- All class methods modeled as `public static` methods
  - Entity passed as the first parameter (except `Log()` and `LogExt()` methods)
- `null` valued variables
  - Start adding "system" fields to each class: `private EntityType _state;`
  - `public static` wrapper methods for all system fields (scoped `private`)
- Use `byte[]` whenever possible to minimize Storage as well as being the easiest to work from a supported runtime library perspective

5. Options (con't)

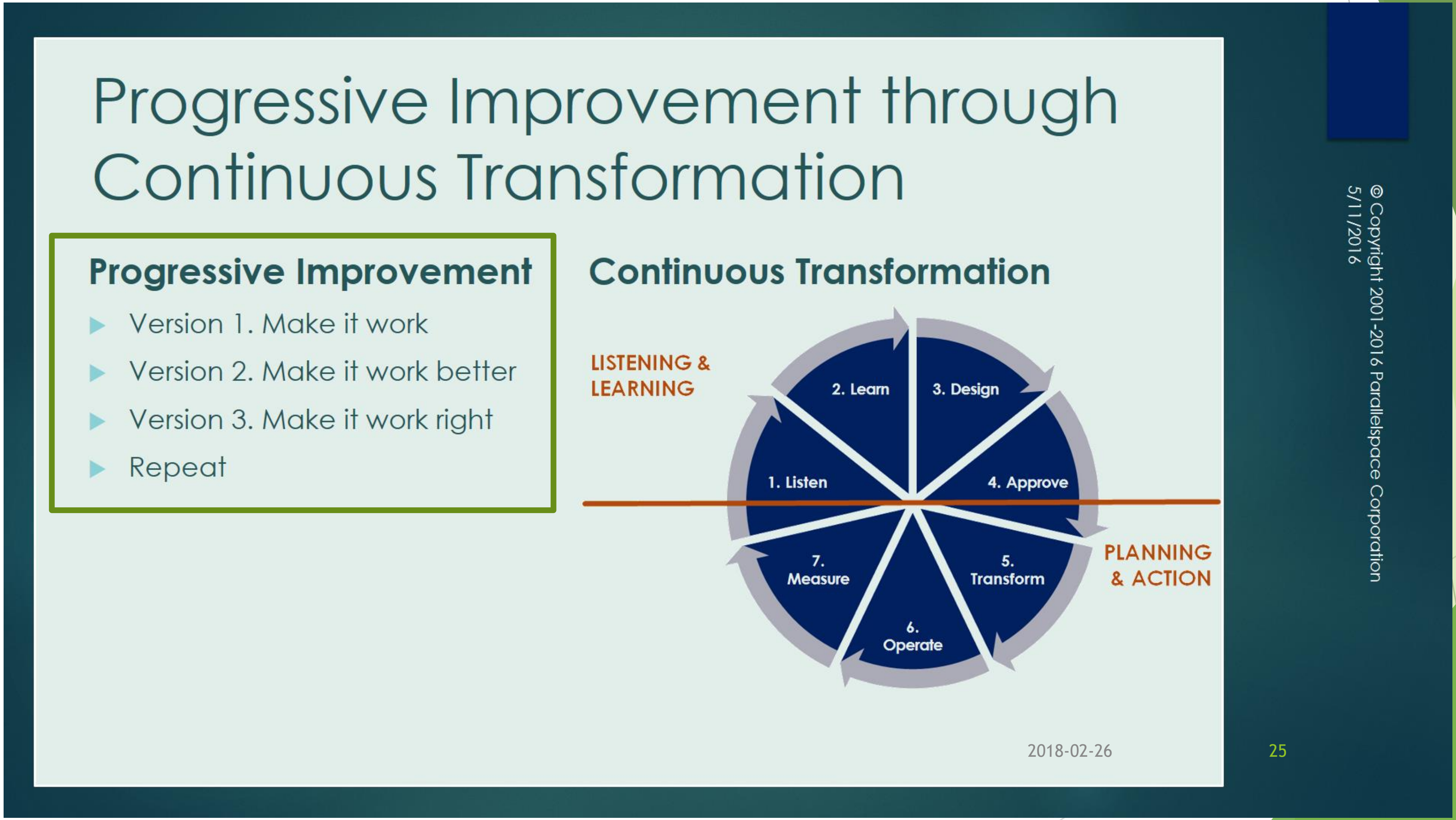
Model every C# Class as an C#.NPC Class with the following characteristics:

- All NPC have the same minimum set of default public static methods with the set of methods varying with the NPC Level being implemented for the class:
  - NPC Level 0 Basic
  - NPC Level 1 Managed
  - NPC Level 2 Persistable
  - NPC Level 3 Deletable (Bury/Tombstone)
  - NPC Level 4 Collectible
  - NPC Level 5 Extendible (Roadmap)
  - NPC Level 6 Authorized (Roadmap)
  - NPC Level 6 Optimized (Roadmap)
- NOTE: Primary driver for NPC Levels is minimizing code size by implementing only the Levels (and methods) required by the specific class

6. Solution

Sidebar: How we think about how we work?

https://hyperonomy.com/2016/05/09/how-do-we-think-about-how-we-work/



6. Solution

- NPC Level 0 Basic
- NPC Level 1 Managed
- NPC Level 2 Persistable
- NPC Level 3 Deletable (Bury/Tombstone)
- NPC Level 4 Collectible
- NPC Level 5 Extendible (Roadmap)
- NPC Level 6 Authorized (Roadmap)
- NPC Level 6 Optimized (Roadmap)

6. Solution – NPC Level 0 Basic

NPC Level 0 Basic - Point

```
public class Point /* Level 0 */
{
    public BigInteger x;
    public BigInteger y;
}
```

6. Solution – NPC Level 1 Managed

NPC Level 1 Managed – NeoTrace

```
public class NeoTrace /* Level *all* */
{
    public static void Trace(params object[] args)
    {
        Neo.SmartContract.Framework.Services.Neo.Runtime.Notify(args);
    }
}
```

NPC Level 1 Managed – NeoEntityModel

```
public static class NeoEntityModel /* Level 1 */
{
    public enum EntityState
    {
        NULL, INIT, SET
    }
    public static BigInteger AsBigInteger(this EntityState sta)
    {
        int istate = (int)sta;
        BigInteger bis = istate;
        return bis;
    }
    public static EntityState BytesToEntityState(byte[] bsta)
    {
        int ista = (int)bsta.AsBigInteger();
        NeoEntityModel.EntityState sta = (NeoEntityModel.EntityState)ista;
        return sta;
    }
}
```

NPC Level 1 Managed – Point

```
public class Point : NeoTrace /* Level 1 */
{
   private BigInteger _x;
   private BigInteger _y;
   private NeoEntityModel.EntityState _state;

   // Accessors
   public static void SetX(Point p, BigInteger value)
          { p._x = value; p._state = NeoEntityModel.EntityState.SET; }
   public static BigInteger GetX(Point p) { return p._x; }
   public static void SetY(Point p, BigInteger value)
          { p._y = value; p._state = NeoEntityModel.EntityState.SET; }
   public static BigInteger GetY(Point p) { return p._y; }
   public static void Set(Point p, BigInteger xvalue, BigInteger yvalue)
          { p._x = xvalue; p._y = yvalue; p._state = NeoEntityModel.EntityState.SET; }
```

NPC Level 1 Managed – Point (con’t)

```
// Factory methods
private Point()
{
}

private static Point _Initialize(Point p)
{
    p._x = 0;
    p._y = 0;
    p._state = NeoEntityModel.EntityState.NULL;
    LogExt("_Initialize(p).p", p);
    return p;
}
public static Point New()
{
    Point p = new Point();
    _Initialize(p);
    LogExt("New().p", p);
    return p;
}
```

NPC Level 1 Managed – Point (con't)

```
public static Point New(int x, int y)
{
    Point p = new Point();
    p._x = x;
    p._y = y;
    p._state = NeoEntityModel.EntityState.INIT;
    LogExt("New(x,y).p", p);
    return p;
}
public static Point Null()
{
    Point p = new Point();
    _Initialize(p);
    LogExt("Null().p", p);
    return p;
}
```

NPC Level 1 Managed – Point (con’t)

```
    // EntityState wrapper methods
    public static bool IsNull(Point p)
    {
        return (p._state == NeoEntityModel.EntityState.NULL);
    }

    // Log/trace methods
    public static void Log(string label, Point p)
    {
        NeoTrace.Trace(label, p._x, p._y);
    }
    public static void LogExt(string label, Point p)
    {
        NeoTrace.Trace(label, p._x, p._y, p._state);
    }
}
```

6. Solution – NPC Level 2 Persistable

NPC Level 2 Persistable - NeoEntityModel

```
public static class NeoEntityModel /* Level 2 Persistable */
{
    public enum EntityState
    {
        NULL, INIT, SET, PUTTED, GETTED, MISSING
    }

    public static BigInteger AsBigInteger(this EntityState state)
    {
        int istate = (int)state;
        BigInteger bis = istate;
        return bis;
    }
    public static EntityState BytesToEntityState(byte[] bsta)
    {
        int ista = (int)bsta.AsBigInteger();
        NeoEntityModel.EntityState sta = (NeoEntityModel.EntityState)ista;
        return sta;
    }
}
```

NPC Level 2 Persistable – Point

```
public class Point : NeoTrace /* Level 2 Persistable */
{
    private BigInteger _x;
    private BigInteger _y;
    private NeoEntityModel.EntityState _state;

    // Accessors
    public static void SetX(Point p, BigInteger value)
           { p._x = value; p._state = NeoEntityModel.EntityState.SET; }
    public static BigInteger GetX(Point p) { return p._x; }
    public static void SetY(Point p, BigInteger value)
           { p._y = value; p._state = NeoEntityModel.EntityState.SET; }
    public static BigInteger GetY(Point p) { return p._y; }
    public static void Set(Point p, BigInteger xvalue, BigInteger yvalue)
           { p._x = xvalue; p._y = yvalue; p._state = NeoEntityModel.EntityState.SET; }
```

NPC Level 2 Persistable – Point (con't)

```
// Class name and property names
private const string _className = "Point";
private const string _sX = "X";
private const string _sY = "Y";
private const string _sSTA = "_STA";
private const string _sEXT = "_EXT";
private static readonly byte[] _bX = Helper.AsByteArray(_sX);
private static readonly byte[] _bY = Helper.AsByteArray(_sY);
private static readonly byte[] _bSTA = Helper.AsByteArray(_sSTA);
private static readonly byte[] _bEXT = Helper.AsByteArray(_sEXT);

// Internal fields
private const string _classKeyTag = "/#" + _className + ".";
private static readonly byte[] _bclassKeyTag = Helper.AsByteArray(_classKeyTag);
```

NPC Level 2 Persistable – Point (con’t)

```
// Factory methods
private Point()
{
}
private static Point _Initialize(Point p)
{
    p._x = 0; p._y = 0;
    p._state = NeoEntityModel.EntityState.NULL;
    LogExt("_Initialize(p).p", p);
    return p;
}
public static Point New()
{
       Point p = new Point();
    _Initialize(p);
    LogExt("New().p", p);
    return p;
}
```

NPC Level 2 Persistable – Point (con’t)

```
public static Point New(int x, int y)
{
    Point p = new Point();
    p._x = x; p._y = y;
    p._state = NeoEntityModel.EntityState.INIT;
    LogExt("New(x,y).p", p);
    return p;
}
public static Point Null()
{
    Point p = new Point();
    _Initialize(p);
    LogExt("Null().p", p);
  return p;
}
```

NPC Level 2 Persistable – Point (con't)

```
// EntityState wrapper methods
public static bool IsNull(Point p)
{
    return (p._state == NeoEntityModel.EntityState.NULL);
}

// Log/trace methods
public static void Log(string label, Point p)
{
    NeoTrace.Trace(label, p._x, p._y);
}
public static void LogExt(string label, Point p)
{
    NeoTrace.Trace(label, p._x, p._y, p._state);
}
```

NPC Level 2 Persistable – Point (con’t)

```
// Persistable methods
public static bool IsMissing(Point p)
{
    return (p._state == NeoEntityModel.EntityState.MISSING);
}

public static Point Missing()
{
    Point p = new Point();
    p._x = 0; p._y = 0;
    p._state = NeoEntityModel.EntityState.MISSING;
    LogExt("Missing().p", p);
    return p;
}
```

NPC Level 2 Persistable – Point (con’t)

```
public static bool Put(Point p, byte[] key)
{
    if (key.Length == 0) return false;
    Neo.SmartContract.Framework.Services.Neo.StorageContext ctx =
            Neo.SmartContract.Framework.Services.Neo.Storage.CurrentContext;
    byte[] _bkeyTag = Helper.Concat(key, _bclassKeyTag);

    p._state = NeoEntityModel.EntityState.PUTTED;
    Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx,
            Helper.Concat(_bkeyTag, _bSTA), p._state.AsBigInteger());
    Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx,
            Helper.Concat(_bkeyTag, _bX), p._x);
    Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx,
            Helper.Concat(_bkeyTag, _bY), p._y);
    LogExt("Put(bkey).p", p);
    return true;
}
```

NPC Level 2 Persistable – Point (con’t)

```
public static Point Get(byte[] key)
{
    if (key.Length == 0) return Null();
    Neo.SmartContract.Framework.Services.Neo.StorageContext ctx =
        Neo.SmartContract.Framework.Services.Neo.Storage.CurrentContext;
    byte[] _bkeyTag = Helper.Concat(key, _bclassKeyTag);
    Point p;
    byte[] bsta = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx, Helper.Concat(_bkeyTag, _bSTA));
    NeoTrace.Trace("Get(kb).bs", bsta.Length, bsta);
    if (bsta.Length == 0)
    {
        p = Point.Missing();
    }
    else // not MISSING
    {
        byte[] bext = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx, Helper.Concat(_bkeyTag, _bEXT));
        int ista = (int)bsta.AsBigInteger();
        NeoEntityModel.EntityState sta = (NeoEntityModel.EntityState)ista;
        p = new Point();
        BigInteger x = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx,
                       Helper.Concat(_bkeyTag, _bX)).AsBigInteger();
        BigInteger y = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx,
                       Helper.Concat(_bkeyTag, _bY)).AsBigInteger();
        p._x = x; p._y = y; p._state = sta;
        p._state = NeoEntityModel.EntityState.GETTED; /* OVERRIDE */
    }
    LogExt("Get(kb).p", p);
    return p;
}
```

NPC Level 2 Persistable – Point (con’t)

```
public static Point Get(byte[] key)
{
    if (key.Length == 0) return Null();
    Neo.SmartContract.Framework.Services.Neo.StorageContext ctx =
        Neo.SmartContract.Framework.Services.Neo.Storage.CurrentContext;
    byte[] _bkeyTag = Helper.Concat(key, _bclassKeyTag);
    Point p;
    byte[] bsta = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx, Helper.Concat(_bkeyTag, _bSTA));
    NeoTrace.Trace("Get(kb).bs", bsta.Length, bsta);
    if (bsta.Length == 0)
    {
        p = Point.Missing();
    }
    else // not MISSING
    {
        byte[] bext = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx, Helper.Concat(_bkeyTag, _bEXT));
        int ista = (int)bsta.AsBigInteger();
        NeoEntityModel.EntityState sta = (NeoEntityModel.EntityState)ista;
        p = new Point();
        BigInteger x = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx,
                           Helper.Concat(_bkeyTag, _bX)).AsBigInteger();
        BigInteger y = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx,
                           Helper.Concat(_bkeyTag, _bY)).AsBigInteger();
        p._x = x; p._y = y; p._state = sta;
        p._state = NeoEntityModel.EntityState.GETTED; /* OVERRIDE */
    }
    LogExt("Get(kb).p", p);
    return p;
}
```

6. Solution – NPC Level 3 Deletable

NPC Level 3 Deletable - NeoEntityModel

```
public static class NeoEntityModel /* Level 3 Deletable */
{
    public enum EntityState
    {
        NULL, INIT, SET, PUTTED, GETTED, MISSING, TOMBSTONED
    }

    public static BigInteger AsBigInteger(this EntityState state)
    {
        int istate = (int)state;
        BigInteger bis = istate;
        return bis;
    }
    public static EntityState BytesToEntityState(byte[] bsta)
    {
        int ista = (int)bsta.AsBigInteger();
        NeoEntityModel.EntityState sta = (NeoEntityModel.EntityState)ista;
        return sta;
    }
}
```

NPC Level 3 Deletable – Point (con't)

```
...
    // Deletable methods
    public static bool IsBuried(Point p)
    {
        return (p._state == NeoEntityModel.EntityState.TOMBSTONED);
    }

    public static Point Tombstone()
    {
        Point p = new Point();
        p._x = 0;
        p._y = 0;
        p._state = NeoEntityModel.EntityState.TOMBSTONED;
        LogExt("Tombstone().p", p);
        return p;
    }
...
```

NPC Level 3 Deletable – Point (con't)

```
...
    public static Point Bury(byte[] key)
    {
       if (key.Length == 0) return Null();

       Neo.SmartContract.Framework.Services.Neo.StorageContext ctx =
                    Neo.SmartContract.Framework.Services.Neo.Storage.CurrentContext;
       byte[] _bkeyTag = Helper.Concat(key, _bclassKeyTag);

       Point p;
       byte[] bsta = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx, Helper.Concat(_bkeyTag, _bSTA));
       NeoTrace.Trace("Bury(kb).bs", bsta.Length, bsta);
       if (bsta.Length == 0)
       {
           p = Point.Missing();
       }
       else // not MISSING - bury it
       {
           p = Point.Tombstone();
           Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx,
                    Helper.Concat(_bkeyTag, _bSTA), p._state.AsBigInteger());
           Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx, Helper.Concat(_bkeyTag, _bX), p._x);
           Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx, Helper.Concat(_bkeyTag, _bY), p._y);
       }
       LogExt("Bury(kb).p", p);
       return p; // return Point p to signal if key is Missing or bad key
    }
...
```

6. Solution – NPC Level 4 Collectible

NPC Level 4 Collectible - NeoEntityModel

```
public static class NeoEntityModel /* Level 4 Collectible */
{
    public enum EntityState
    {
        NULL, INIT, SET, PUTTED, GETTED, MISSING, TOMBSTONED, NOTAUTHORIZED /* Future */
    }
    public static BigInteger AsBigInteger(this EntityState state)
    {
        int istate = (int)state;
        BigInteger bis = istate;
        return bis;
    }
    public static EntityState BytesToEntityState(byte[] bsta)
    {
        int ista = (int)bsta.AsBigInteger();
        NeoEntityModel.EntityState sta = (NeoEntityModel.EntityState)ista;
        return sta;
    }
    public static readonly byte[] NullScriptHash = "".ToScriptHash();
    public static readonly byte[] NullByteArray = "".AsByteArray();
```

NPC Level 4 Collectible - NeoVersionedAppUser

```
public class NeoVersionedAppUser
{
    private byte[] _app;
    private int _major;
    private int _minor;
    private int _build;
    private byte[] _userScriptHash;
    private NeoEntityModel.EntityState _state;
```

NPC Level 4 Collectible - NeoVersionedAppUser

```
public static void SetAppName(NeoVersionedAppUser vau, byte[] value)
                          { vau._app = value; vau._state = NeoEntityModel.EntityState.SET; }
public static byte[] GetAppNameAsByteArray(NeoVersionedAppUser vau) { return vau._app; }
public static void SetAppName(NeoVersionedAppUser vau, string value)
                     { vau._app = value.AsByteArray(); vau._state = NeoEntityModel.EntityState.SET; }
public static string GetAppNameAsString(NeoVersionedAppUser vau) { return vau._app.AsString(); }
public static void SetMajor(NeoVersionedAppUser vau, int value)
                     { vau._major = value; vau._state = NeoEntityModel.EntityState.SET; }
public static int GetMajor(NeoVersionedAppUser vau) { return vau._major; }
public static void SetMinor(NeoVersionedAppUser vau, int value)
                     { vau._minor = value; vau._state = NeoEntityModel.EntityState.SET; }
public static int GetMinor(NeoVersionedAppUser vau) { return vau._minor; }
public static void SetBuild(NeoVersionedAppUser vau, int value)
                     { vau._build = value; vau._state = NeoEntityModel.EntityState.SET; }
public static int GetBuild(NeoVersionedAppUser vau) { return vau._build; }
public static void SetUserScriptHash(NeoVersionedAppUser vau, byte[] value)
                     { vau._userScriptHash = value; vau._state = NeoEntityModel.EntityState.SET; }
public static byte[] GetUserScriptHash(NeoVersionedAppUser vau) { return vau._userScriptHash; }
public static void Set(NeoVersionedAppUser vau, byte[] app, int major, int minor, int build, byte[] userScriptHash)
{
    vau._app = app; vau._major = major; vau._minor = minor; vau._build = build;
    vau._userScriptHash = userScriptHash; vau._state = NeoEntityModel.EntityState.SET;
}
public static void Set(NeoVersionedAppUser vau, string app, int major, int minor, int build, byte[] userScriptHash)
{
    vau._app = app.AsByteArray(); vau._major = major; vau._minor = minor; vau._build = build;
    vau._userScriptHash = userScriptHash; vau._state = NeoEntityModel.EntityState.SET;
}
```

NPC Level 4 Collectible - NeoStorageKey

```
public class NeoStorageKey
{
    private byte[] _app;
    private int _major;
    private int _minor;
    private int _build;
    private byte[] _userScriptHash;
    private byte[] _className;
    private int _index;
    private string _fieldName;
    private NeoEntityModel.EntityState _state;
```

NPC Level 4 Collectible - NeoStorageKey (con't)

```
public static NeoStorageKey New(NeoVersionedAppUser vau, byte[] className)
{
    if (NeoVersionedAppUser.IsNull(vau))
    {
        return NeoStorageKey.Null();
    }

    NeoStorageKey nsk = new NeoStorageKey();
    nsk._app = NeoVersionedAppUser.GetAppNameAsByteArray(vau);
    nsk._major = NeoVersionedAppUser.GetMajor(vau);
    nsk._minor = NeoVersionedAppUser.GetMinor(vau);
    nsk._build = NeoVersionedAppUser.GetBuild(vau);
    nsk._userScriptHash = NeoVersionedAppUser.GetUserScriptHash(vau);
    nsk._className = className;
    nsk._index = 0;
    nsk._fieldName = "";
    nsk._state = NeoEntityModel.EntityState.INIT;
    LogExt("New(vau,bc)", nsk);
    return nsk;
}
```

NPC Level 4 Collectible - NeoStorageKey (con't)

```
private static readonly byte[] _bLeftBrace = "{".AsByteArray();
private static readonly byte[] _bRightBrace = "}".AsByteArray();
private static readonly byte[] _bColon = ":".AsByteArray();
private static readonly byte[] _bEquals = "=".AsByteArray();
private static readonly byte[] _bSemiColon = ";".AsByteArray();
private static readonly byte[] _ba = "a".AsByteArray(); // App name
private static readonly byte[] _bM = "M".AsByteArray(); // App major version
private static readonly byte[] _bm = "m".AsByteArray(); // App minor version
private static readonly byte[] _bb = "b".AsByteArray(); // App build number
private static readonly byte[] _bu = "u".AsByteArray(); // User script hash
private static readonly byte[] _bc = "c".AsByteArray(); // Class name
private static readonly byte[] _bi = "i".AsByteArray(); // Index value
private static readonly byte[] _bf = "f".AsByteArray(); // Field name

private static readonly byte[] _bStringType =
                               { (byte)Neo.SmartContract.ContractParameterType.String };
private static readonly byte[] _bBigIntegerType =
                               { (byte)Neo.SmartContract.ContractParameterType.Integer };
private static readonly byte[] _bUserScriptHashType =
                               { (byte)Neo.SmartContract.ContractParameterType.ByteArray };
```

NPC Level 4 Collectible - NeoStorageKey (con't)

```
public static byte[] StorageKey(NeoStorageKey nsk, int index, byte[] fieldName)
{
    LogExt("StorageKey(nsk,i,fb).nsk", nsk);
    byte[] bkey = Helper.Concat(_bLeftBrace, _ba).Concat(_bColon).Concat(_bStringType)
                .Concat(_bEquals).Concat(nsk._app).Concat(_bSemiColon);
    bkey = Helper.Concat(bkey, _bM).Concat(_bColon).Concat(_bBigIntegerType)
                .Concat(_bEquals).Concat(((BigInteger)(nsk._major)).AsByteArray()).Concat(_bSemiColon);
    bkey = Helper.Concat(bkey, _bm).Concat(_bColon).Concat(_bBigIntegerType)
                .Concat(_bEquals).Concat(((BigInteger)(nsk._minor)).AsByteArray()).Concat(_bSemiColon);
    bkey = Helper.Concat(bkey, _bb).Concat(_bColon).Concat(_bBigIntegerType)
                .Concat(_bEquals).Concat(((BigInteger)(nsk._build)).AsByteArray()).Concat(_bSemiColon);
   bkey = Helper.Concat(bkey, _bu).Concat(_bColon).Concat(_bUserScriptHashType)
                .Concat(_bEquals).Concat(nsk._userScriptHash).Concat(_bSemiColon);
    bkey = Helper.Concat(bkey, _bc).Concat(_bColon).Concat(_bStringType)
                .Concat(_bEquals).Concat(nsk._className).Concat(_bSemiColon);

    bkey = Helper.Concat(bkey, _bi).Concat(_bColon).Concat(_bBigIntegerType)
                .Concat(_bEquals).Concat(((BigInteger)(index)).AsByteArray()).Concat(_bSemiColon);
    bkey = Helper.Concat(bkey, _bf).Concat(_bColon).Concat(_bStringType)
                .Concat(_bEquals).Concat(fieldName).Concat(_bSemiColon);
    bkey = Helper.Concat(bkey, _bRightBrace);
    NeoTrace.Trace("StorageKey(nsk).bkey$BSK", bkey);
    return bkey;
}
```

NPC Level 4 Collectible - Point

```
public static Point BuryElement(NeoVersionedAppUser vau, int index)
{
    if (NeoVersionedAppUser.IsNull(vau)) return Point.Null();

    Neo.SmartContract.Framework.Services.Neo.StorageContext ctx =
                            Neo.SmartContract.Framework.Services.Neo.Storage.CurrentContext;
    NeoStorageKey nsk = NeoStorageKey.New(vau, "Point");
    byte[] bkey;
    Point p;
    byte[] bsta = Neo.SmartContract.Framework.Services.Neo.Storage.Get(ctx, NeoStorageKey.StorageKey(nsk, index, _bSTA));
    NeoTrace.Trace("Bury(vau,index).bs", bsta.Length, bsta);
    if (bsta.Length == 0)
    {
        p = Point.Missing();
    }
    else // not MISSING - bury it
    {
        p = Point.Tombstone();
        Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx, NeoStorageKey.StorageKey(nsk, index, _bSTA),
                                                              p._state.AsBigInteger());
        Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx, NeoStorageKey.StorageKey(nsk, index, _bEXT),
                                                              p._extension);
        Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx, NeoStorageKey.StorageKey(nsk, index, _bX), p._x);
        Neo.SmartContract.Framework.Services.Neo.Storage.Put(ctx, NeoStorageKey.StorageKey(nsk, index, _bY), p._y);
    }
    LogExt("Bury(vau,i).p", p);
    return p;
}
```

7. Test Results

7. Test Results

- `Test1()`
  - Dump miscellaneous variables to the log
- `Test2()`
  - NPC Level 1 test cases: Create 3 Points and a line. Add two Points. Log the results.
- `Test3()`
  - NPC Level 2/3 test cases: Create 3 Points and test NPC Level 3 entity persistence. Log the results.
- `Test4()`
  - NPC Level 2/3 test cases: Test IsNull(), IsMissing() and IsExtended(). Log the results.
- `Test5()`
  - NPC Level 4 test cases: Test NeoStorageKeys. Log the results.
- `Test6()`
  - NPC Level 4 test cases: Test NeoStorageKeys. Log the results

Test1() - Code

```
public static string test1(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("NullHash", NeoEntityModel.NullScriptHash);

    NeoTrace.Trace("NeoEntityModel.EntityState...");
    NeoEntityModel.EntityState state1 = NeoEntityModel.EntityState.MISSING;
    NeoTrace.Trace("state", state1);
    int istate = (int)state1;
    NeoTrace.Trace("state1", state1);

    BigInteger bis = state1.AsBigInteger();
    NeoTrace.Trace("bis", bis);

    byte[] bsta = { 0x4 };
    NeoTrace.Trace("bsta", bsta);
    NeoEntityModel.EntityState state2 = NeoEntityModel.BytesToEntityState(bsta);
    NeoTrace.Trace("state2", state2);

    return msg;
}
```

Test1() - neo-gui Event Log

neo-gui - ChainDockerPrivateNet2

Wallet Transaction Advanced Help

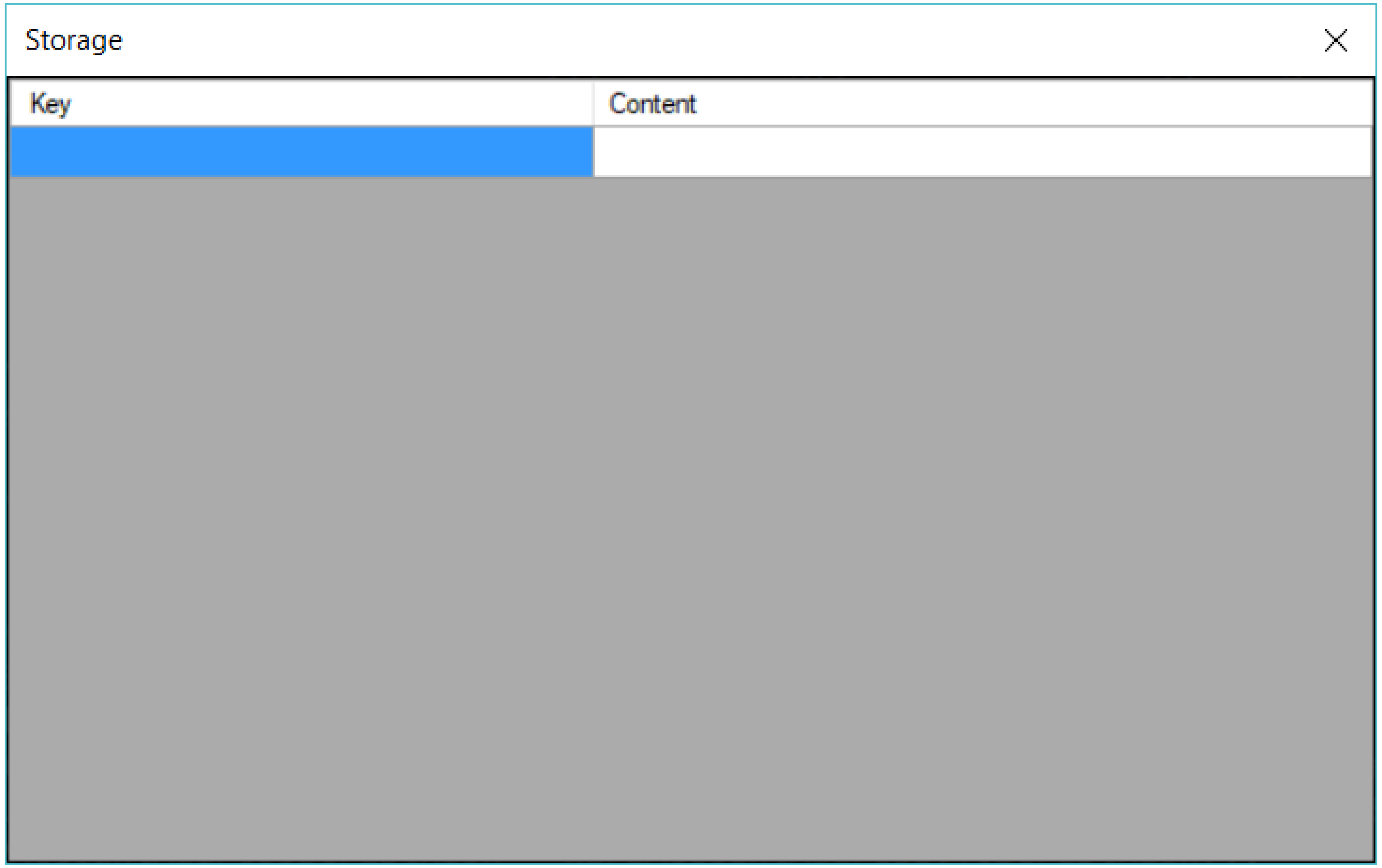
Account Asset Transaction History Event Log

Time	Block	Script Hash	Name	Type	Message
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NPCdApp - NEO Persistable Class (NPC) Framework
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NPCdApp - Version 0.1 Reference Implementation
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	operation / (STR5) 7465737431 'test1' /
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NullHash / (BIN0) [binary]
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NeoEntityModel.EntityState...
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	state / (BINT1) 5
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	state1 / (BINT1) 5
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	bis / (BINT1) 5
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	bsta / (STR1) 04 '□'
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	state2 / (STR1) 04 '□'
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NPCdApp - NEO Persistable Class (NPC) Framework
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NPCdApp - Version 0.1 Reference Implementation
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	operation / (STR5) 7465737431 'test1' /
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NullHash / (BIN0) [binary]
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	NeoEntityModel.EntityState...
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	state / (BINT1) 5
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	state1 / (BINT1) 5
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	bis / (BINT1) 5
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	bsta / (STR1) 04 '□'
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	state2 / (STR1) 04 '□'
2018-02-2...	993	0x7074acf3f06dd3...	a	Notify	--

Height: 997/997/997 Connected: 3

Waiting for next block:

Test1() – neo-debugger Storage Viewer



Test2() - Code

```
public static string test2(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Make P0...");
    Point p0 = Point.New();
    Point.Log("p0", p0);
    Point.SetX(p0, 7);
    Point.SetY(p0, 8);
    Point.Log("p0", p0);
    Point.Set(p0, 9, 10);
    Point.Log("p0", p0);

    NeoTrace.Trace("Make P1...");
    Point p1 = Point.New();
    Point.Set(p1, 2, 4);
    Point.Log("p1", p1);

    NeoTrace.Trace("Make P2...");
    Point p2 = Point.New();
    Point.Set(p2, 15, 16);
    Point.Log("p2", p2);

    NeoTrace.Trace("Make line1...");
    Point[] line1 = new[]
    {
        p1, p2
    };
    NeoTrace.Trace("line1", line1, p1, p2); // TODO: neo-gui doesn't understand this: line1

    NeoTrace.Trace("Add 2 points...");
    Point p3 = Add(line1[0], line1[1]);
    Point.Log("p3", p3);

    return msg;
}
```

Test2() - neo-gui Event Log

neo-gui - ChainDockerPrivateNet2

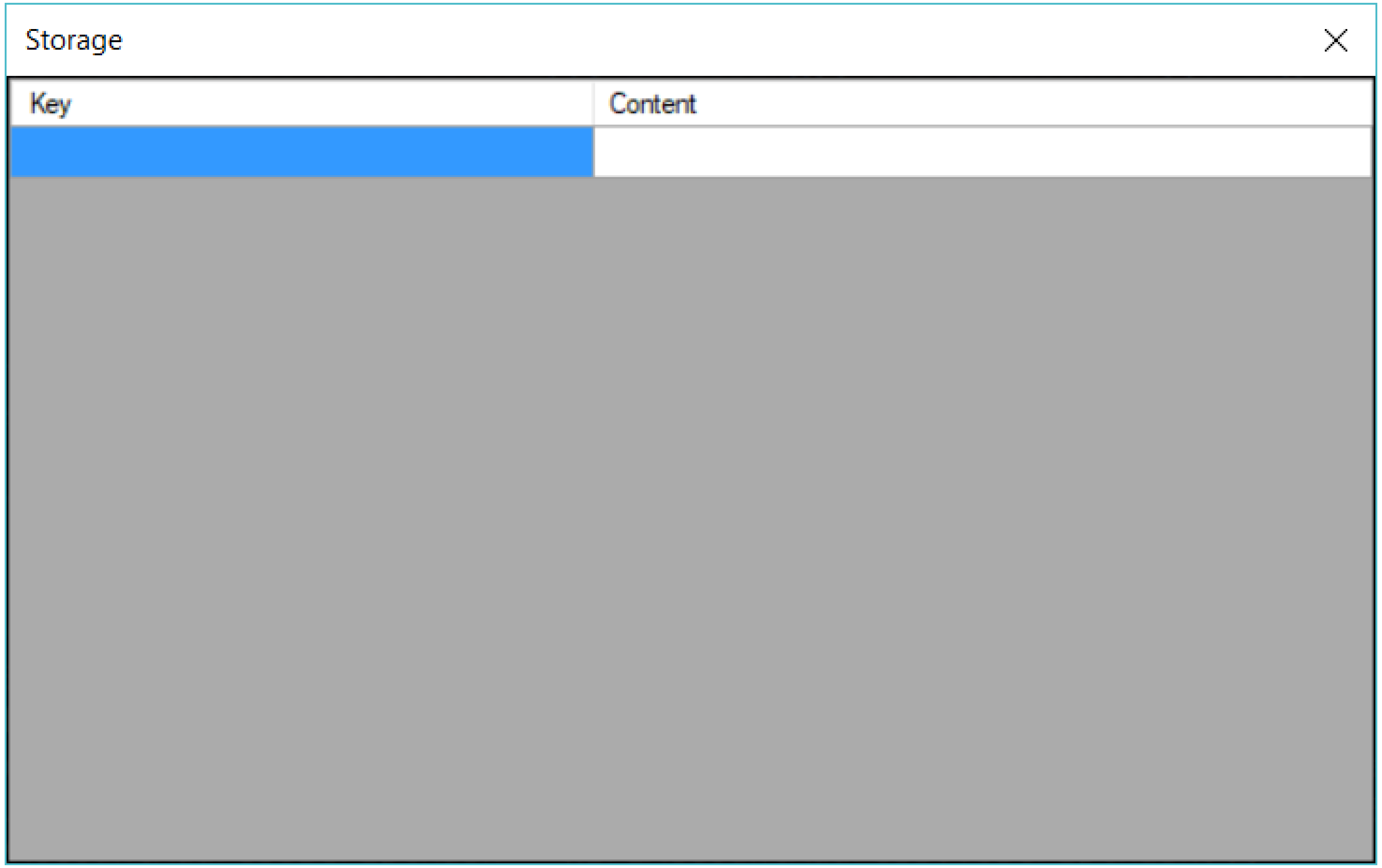
Wallet Transaction Advanced Help

Account Asset Transaction History Event Log

Time	Block	Script Hash	Name	Type	Message
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	line1 / / /
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	Add 2 points...
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	p3 / (BINT2) 17 / (BINT2) 20
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	NPCdApp - NEO Persistable Class (NPC) Framework
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	NPCdApp - Version 0.1 Reference Implementation
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	operation / (STR5) 7465737432 'test2' /
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	Make P0...
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	p0 / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	p0 / (BINT1) 7 / (BINT1) 8
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	p0 / (BINT1) 9 / (BINT2) 10
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	Make P1...
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	p1 / (BINT1) 2 / (BINT1) 4
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	Make P2...
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	p2 / (BINT2) 15 / (BINT2) 16
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	Make line1...
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	line1 / / /
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	Add 2 points...
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	p3 / (BINT2) 17 / (BINT2) 20
2018-02-2...	1002	0x7074acf3f06dd3...	a	Notify	--

Height: 1003/1003/1003 Connected: 2 Waiting for next block:

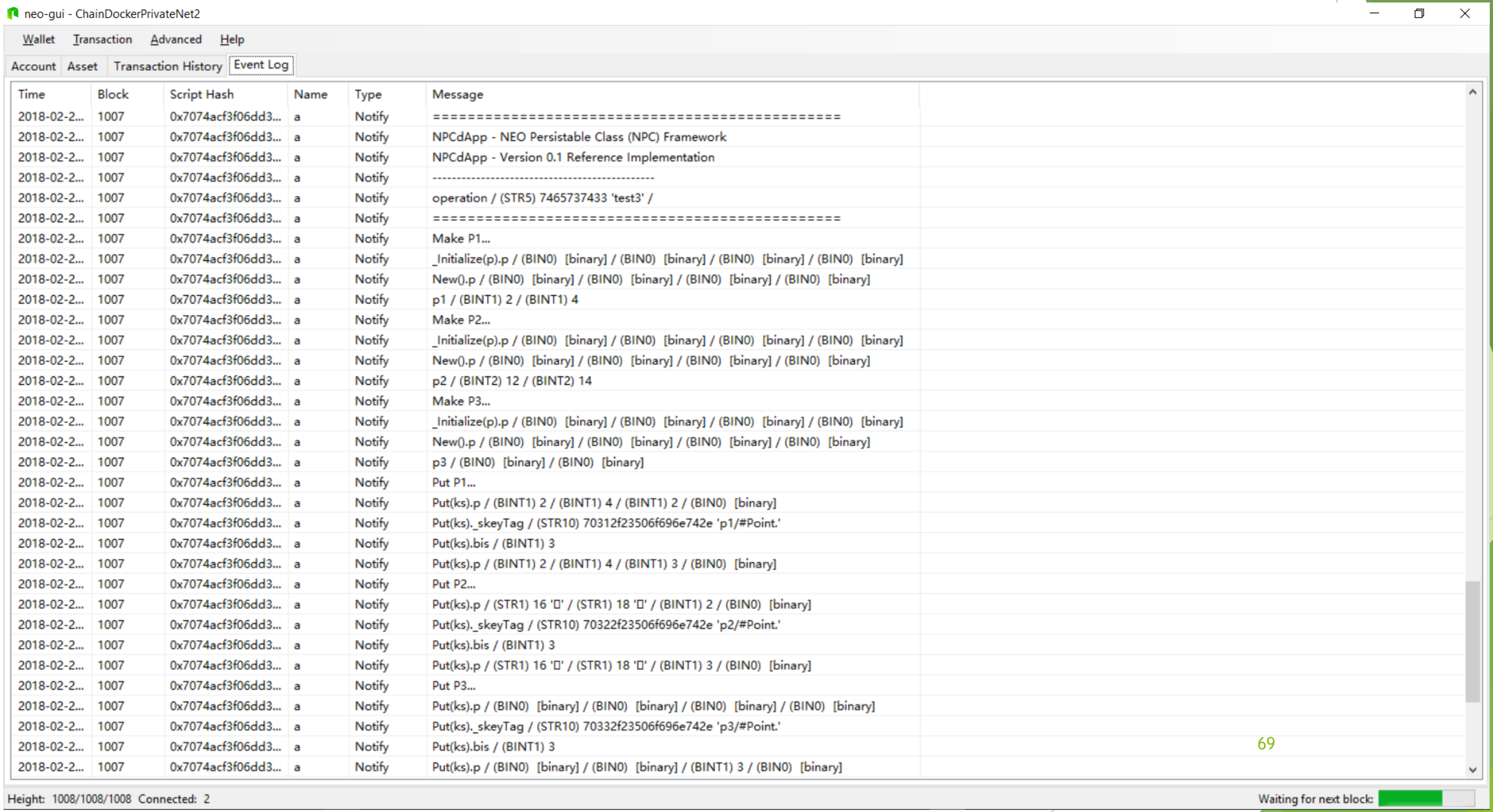
Test2() – neo-debugger Storage Viewer



Test3() - Code

```
public static string test3(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Make P1...");
    Point p1 = Point.New();
    Point.Set(p1, 2, 4);
    Point.Log("p1", p1);

    NeoTrace.Trace("Make P2...");
    Point p2 = Point.New();
    Point.Set(p2, 12, 14);
    Point.Log("p2", p2);

    NeoTrace.Trace("Make P3...");
    Point p3 = Point.New();
    Point.Set(p2, 22, 24);
    Point.Log("p3", p3);

    NeoTrace.Trace("Put P1...");
    Point.Put(p1, "p1");
    NeoTrace.Trace("Put P2...");
    Point.Put(p2, "p2");
    NeoTrace.Trace("Put P3...");
    Point.Put(p3, "p3");

    NeoTrace.Trace("Get P1...");
    Point p1get = Point.Get("p1");
    Point.Log("p1get", p1get);
    NeoTrace.Trace("Get P2...");
    Point p2get = Point.Get("p2");
    Point.Log("p2get", p2get);
    NeoTrace.Trace("Get P3...");
    Point p3get = Point.Get("p3");
    Point.Log("p3get", p3get);

    return msg;
}
```

Test3() – neo-gui Event Log



Time	Block	Script Hash	Name	Type	Message
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	NPCdApp - NEO Persistable Class (NPC) Framework
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	NPCdApp - Version 0.1 Reference Implementation
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	operation / (STR5) 7465737433 'test3' /
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Make P1...
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	p1 / (BINT1) 2 / (BINT1) 4
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Make P2...
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	p2 / (BINT2) 12 / (BINT2) 14
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Make P3...
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	p3 / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put P1...
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).p / (BINT1) 2 / (BINT1) 4 / (BINT1) 2 / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks)._skeyTag / (STR10) 70312f23506f696e742e 'p1/#Point.'
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).bis / (BINT1) 3
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).p / (BINT1) 2 / (BINT1) 4 / (BINT1) 3 / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put P2...
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).p / (STR1) 16 '□' / (STR1) 18 '□' / (BINT1) 2 / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks)._skeyTag / (STR10) 70322f23506f696e742e 'p2/#Point.'
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).bis / (BINT1) 3
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).p / (STR1) 16 '□' / (STR1) 18 '□' / (BINT1) 3 / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put P3...
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks)._skeyTag / (STR10) 70332f23506f696e742e 'p3/#Point.'
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).bis / (BINT1) 3
2018-02-2...	1007	0x7074acf3f06dd3...	a	Notify	Put(ks).p / (BIN0) [binary] / (BIN0) [binary] / (BINT1) 3 / (BIN0) [binary]

Test3() – neo-debugger Storage Viewer

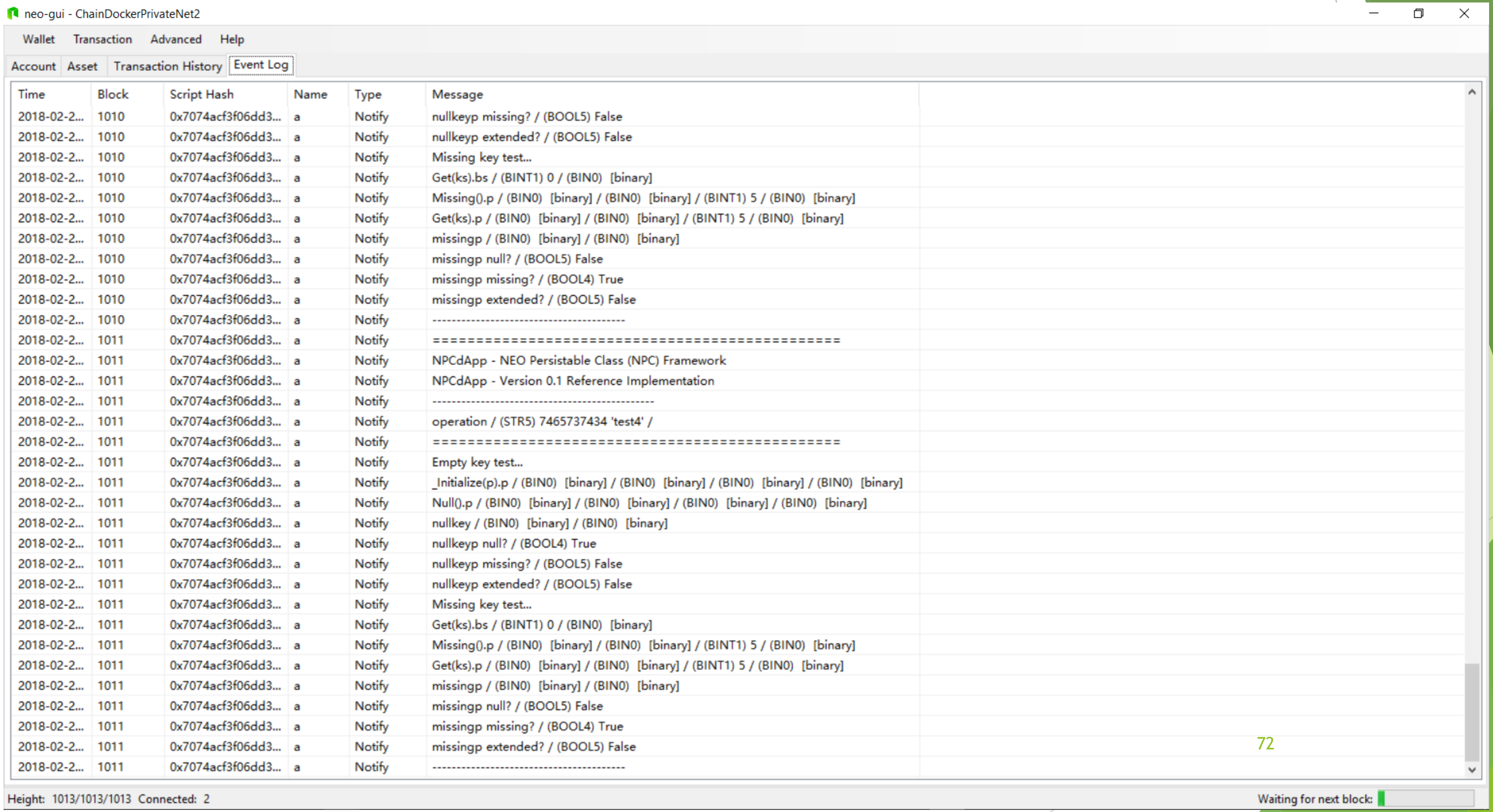
Storage

Key	Content
p1/#Point._STA	3
p1/#Point._EXT	
p1/#Point.X	2
p1/#Point.Y	4
p2/#Point._STA	3
p2/#Point._EXT	
p2/#Point.X	22
p2/#Point.Y	24
p3/#Point._STA	3
p3/#Point._EXT	
p3/#Point.X	
p3/#Point.Y	

Test4() - Code

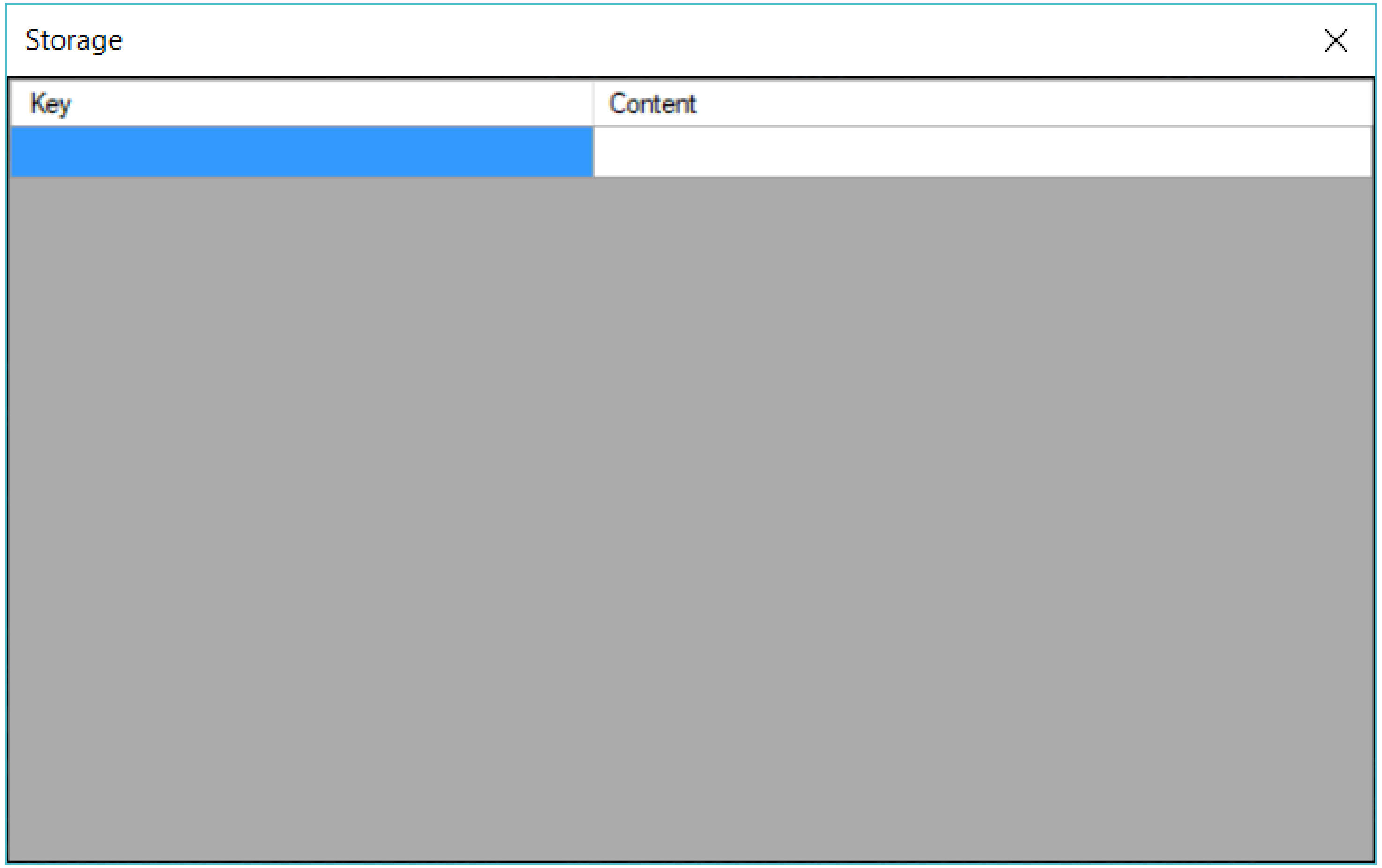
```
public static string test4(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Empty key test...");
    Point nullkeyp = Point.Get("");
    Point.Log("nullkey", nullkeyp);
    NeoTrace.Trace("nullkeyp null?", Point.IsNull(nullkeyp));
    NeoTrace.Trace("nullkeyp missing?", Point.IsMissing(nullkeyp));
    NeoTrace.Trace("nullkeyp extended?", Point.IsExtended(nullkeyp));

    NeoTrace.Trace("Missing key test...");
    Point missingp = Point.Get("missingp");
    Point.Log("missingp", missingp);
    NeoTrace.Trace("missingp null?", Point.IsNull(missingp));
    NeoTrace.Trace("missingp missing?", Point.IsMissing(missingp));
    NeoTrace.Trace("missingp extended?", Point.IsExtended(missingp));

    return msg;
}
```

Test4() – neo-gui Event Log



Time	Block	Script Hash	Name	Type	Message
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	nullkeyp missing? / (BOOL5) False
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	nullkeyp extended? / (BOOL5) False
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	Missing key test...
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	Get(ks).bs / (BINT1) 0 / (BIN0) [binary]
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	Missing().p / (BIN0) [binary] / (BIN0) [binary] / (BINT1) 5 / (BIN0) [binary]
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	Get(ks).p / (BIN0) [binary] / (BIN0) [binary] / (BINT1) 5 / (BIN0) [binary]
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	missingp / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	missingp null? / (BOOL5) False
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	missingp missing? / (BOOL4) True
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	missingp extended? / (BOOL5) False
2018-02-2...	1010	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	===
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	NPCdApp - NEO Persistable Class (NPC) Framework
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	NPCdApp - Version 0.1 Reference Implementation
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	operation / (STR5) 7465737434 'test4' /
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	===
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	Empty key test...
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	Null().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	nullkey / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	nullkeyp null? / (BOOL4) True
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	nullkeyp missing? / (BOOL5) False
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	nullkeyp extended? / (BOOL5) False
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	Missing key test...
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	Get(ks).bs / (BINT1) 0 / (BIN0) [binary]
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	Missing().p / (BIN0) [binary] / (BIN0) [binary] / (BINT1) 5 / (BIN0) [binary]
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	Get(ks).p / (BIN0) [binary] / (BIN0) [binary] / (BINT1) 5 / (BIN0) [binary]
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	missingp / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	missingp null? / (BOOL5) False
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	missingp missing? / (BOOL4) True
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	missingp extended? / (BOOL5) False
2018-02-2...	1011	0x7074acf3f06dd3...	a	Notify	--

Test4() – neo-debugger Storage Viewer



Test5() - Code

```
public static string test5(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Test NeoStorageKeys...");
    Point p4 = Point.New();
    Point.Set(p4, 10, 20);
    Point.Log("p4", p4);

    string app = "FooBar";
    byte[] user = WIF2AccountAddressScriptHash;
    NeoVersionedAppUser vau = NeoVersionedAppUser.New(app, 1, 0, 2034, user);
    NeoVersionedAppUser.Log("test5.vau", vau);

    int index = 24;
    NeoTrace.Trace("index", index);
    Point.PutElement(p4, vau, index);
    index = 25;
    NeoTrace.Trace("index", index);
    Point.PutElement(p4, vau, index);

    index = 24;
    Point p4get = Point.GetElement(vau, index);
    Point.LogExt("p4get", p4get);

    Point p4bury1 = Point.BuryElement(vau, index);
    Point.LogExt("p4bury1", p4bury1);
    Point p4bury2 = Point.GetElement(vau, index);
    Point.LogExt("p4bury2", p4bury2);

    return msg;
}
```

Test5() – neo-gui Event Log

neo-gui - ChainDockerPrivateNet2

Wallet Transaction Advanced Help

Account Asset Transaction History Event Log

Time	Block	Script Hash	Name	Type	Message
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	NPCdApp - NEO Persistable Class (NPC) Framework
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	NPCdApp - Version 0.1 Reference Implementation
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	operation / (STR5) 7465737435 'test5' /
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	Test NeoStorageKeys...
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	p4 / (BINT2) 10 / (STR1) 14 '□'
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	New(a,m,m,b,u).vau / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (ADHASH20) 23ba2703c53263e8d6e522dc32203339dcd8eee9 [binary] / (BINT1) 1
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	test5.vau / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (ADHASH20) 23ba2703c53263e8d6e522dc32203339dcd8eee9 [binary]
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	index / (STR1) 18 '□'
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	New(vau,cs).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23ba2703c
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK80) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=#?'□?2c???"?2 39????;c:□=Point;i: :□;f:□:_STA}
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK80) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=#?'□?2c???"?2 39????;c:□=Point;i: :□;f:□:_EXT}
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK77) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=#?'□?2c???"?2 39????;c:□=Point;i: :□;f:□:X}
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK77) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=#?'□?2c???"?2 39????;c:□=Point;i: :□;f:□:Y}
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	PutElement(vau,i).p / (BINT2) 10 / (STR1) 14 '□' / (BINT1) 3 / (BIN0) [binary]
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	index / (STR1) 19 '□'
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	New(vau,cs).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23ba2703c
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK80) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=#?'□?2c???"?2 39????;c:□=Point;i: :□;f:□:_STA}
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK80) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=#?'□?2c???"?2 39????;c:□=Point;i: :□;f:□:_EXT}
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK77) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=#?'□?2c???"?2 39????;c:□=Point;i: :□;f:□:X}
2018-02-2...	1016	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (ADHASH20) 23b

Height: 1028/1028/1028 Connected: 2

Waiting for next block:

Test5() – neo-debugger Storage Viewer

Storage

Key	Content
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=24;c:7=_STA;}	6
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=24;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=24;c:7=X;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=24;c:7=Y;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=25;c:7=_STA;}	3
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=25;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=25;c:7=X;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=25;c:7=Y;}	20

Test6() - Code

```
public static string test6(object[] args)
{
    string msg = "success";
    int maxIterations = 10;
    if (args.Length > 0)
    {
        maxIterations = (int)((byte[])args[0]).AsBigInteger();
        NeoTrace.Trace("maxIterations", maxIterations);
    }
    if (maxIterations <= 0) maxIterations = 10;
    if (maxIterations > 20) maxIterations = 10;
    NeoTrace.Trace("maxIterations", maxIterations);

    byte[] callingUserScriptHash = ExecutionEngine.CallingScriptHash;
    NeoTrace.Trace("callingUserScriptHash", callingUserScriptHash);
    byte[] entryUserScriptHash = ExecutionEngine.EntryScriptHash;
    NeoTrace.Trace("entryUserScriptHash", entryUserScriptHash);
    byte[] executingUserScriptHash = ExecutionEngine.ExecutingScriptHash;
    NeoTrace.Trace("executingUserScriptHash", executingUserScriptHash);
    byte[] invokingUserScriptHash = GetInvokingUserScriptHash();
    NeoTrace.Trace("invokingUserScriptHash", invokingUserScriptHash.Length, invokingUserScriptHash);
    if (invokingUserScriptHash.Length == 0) invokingUserScriptHash = WIF2AccountAddressScriptHash;
    NeoTrace.Trace("invokingUserScriptHash", invokingUserScriptHash);

    Point p4 = Point.New();
    Point.Set(p4, 10, 20);
    Point.Log("p4", p4);
```

Test6() - Code (con't)

```
    string app = "FooBar";
    NeoVersionedAppUser vau = NeoVersionedAppUser.New(app, 1, 0, 2034, invokingUserScriptHash);
    NeoVersionedAppUser.Log("test6.vau", vau);

    int iteration = 0;
    for (int index = 30; index < 40; index++)
    {
        NeoTrace.Trace("index", index);
        Point.Set(p4, index, -index);
        Point.PutElement(p4, vau, index);
        iteration++;
        if (iteration > maxIterations) break;
    }

    iteration = 0;
    for (int index = 30; index < 40; index++)
    {
        Point.Set(p4, index, -index);
        Point x = Point.GetElement(vau, index);
        Point.Log("loop.x", x);
        if (Point.GetX(p4) != index || Point.GetY(p4) != -index)
        {
            msg = ">>>>(x,y) are different";
            NeoTrace.Trace(msg);
            break;
        }
        iteration++;
        if (iteration > maxIterations) break;
    }

    return msg;
}
```

Test6() – neo-gui Event Log

neo-gui - ChainDockerPrivateNet2

Wallet Transaction Advanced Help

Account Asset Transaction History Event Log

Time	Block	Script Hash	Name	Type	Message
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	NPCdApp - NEO Persistable Class (NPC) Framework
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	NPCdApp - Version 0.1 Reference Implementation
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	--
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	operation / (STR5) 7465737436 'test6' /
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	==
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	maxIterations / (BINT1) 5
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	callingUserScriptHash / (ADHASH20) bc7eb0045b1cf6eb5310e156f4d36df0f3ac7470 [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	entryUserScriptHash / (ADHASH20) 4fe55ee9a25bf0c59f0116cee34fcdb6969aebb4 [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	executingUserScriptHash / (ADHASH20) bc7eb0045b1cf6eb5310e156f4d36df0f3ac7470 [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	invokingUserScriptHash / (BIN0) [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	_Initialize(p).p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	New().p / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	p4 / (BINT2) 10 / (STR1) 14 '□'
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	New(a,m,m,b,u).vau / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (BIN0) [binary] / (BINT1) 1
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	test6.vau / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (BIN0) [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	New(vau,cs).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BINT1
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] ,
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK60) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=;c:□=Point;i: :;f:□:_STA}
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] ,
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK60) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=;c:□=Point;i: :;f:□:_EXT}
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] ,
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK57) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=;c:□=Point;i: :;f:□:X}
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] ,
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK57) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=;c:□=Point;i: :;f:□:Y}
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	PutElement(vau,i).p / (STR1) 1e '' / (BINT3) -30 / (BINT1) 3 / (BIN0) [binary]
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	New(vau,cs).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] / (BINT1
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] ,
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK60) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=;c:□=Point;i: :;f:□:_STA}
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] ,
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk).bkey$BSK / ($BSK60) {a:□=FooBar;M: =□;m: =;b: =?□;u:□=;c:□=Point;i: :;f:□:_EXT}
2018-02-2...	1021	0x7074acf3f06dd3...	a	Notify	StorageKey(nsk,i,fb).nsk / (STR6) 466f6f426172 'FooBar' / (BINT1) 1 / (BIN0) [binary] / (STR2) f207 '?□' / (STR5) 506f696e74 'Point' / (BIN0) [binary] / (BIN0) [binary] / (BIN0) [binary] ,

Height: 1023/1023/1023 Connected: 2

Waiting for next block:

Test6() – neo-debugger Storage Viewer

Storage

Key	Content
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=30;c:7=_STA;}	3
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=30;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=30;c:7=X;}	30
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=30;c:7=Y;}	?
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=31;c:7=_STA;}	3
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=31;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=31;c:7=X;}	31
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=31;c:7=Y;}	?
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=32;c:7=_STA;}	3
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=32;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=32;c:7=X;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=32;c:7=Y;}	?
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=33;c:7=_STA;}	3
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=33;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=33;c:7=X;}	!
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=33;c:7=Y;}	?
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=34;c:7=_STA;}	3
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=34;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=34;c:7=X;}	"
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=34;c:7=Y;}	?
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=35;c:7=_STA;}	3
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=35;c:7=_EXT;}	
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=35;c:7=X;}	#
{a:7=FooBar;M:2=1;m:2=0;b:2=2034;u:5=23-BA-27-03-C5-32-63-E8-D6-E5-22-DC-32-20-33-39-DC-D8-EE-E9;c:7=Point;i:2=35;c:7=Y;}	?

8. Roadmap

8. Roadmap

- Automatic Code Generation
  - In NPC 1.0, Point() was handcoded for each NPC Level but with an eye to being automatically generated in the future
- Add support for Storage.Delete()
  - Simply overlooked in NPC 1.0
- Level 5 Extendible
  - Support entity extension capabilities for existing entities living in deployed smart contracts
- Level 6 Authorized
  - Support for to support authenticated and role-based authorization to entities you create (and others create)
- Level 7 Optimized
  - Key space optimizations starting with NeoStorageKey compression

Roadmap: Automatic Code Generation

- NPC Compiler (npcc.exe)
- Compile .cs file something like the following into a DLL
- Then pass the DLL plus NPC Level of persistence support you need to npcc.exe and a new expanded .cs file is created.

```
namespace NeoPesistenceClasses0
{
    public class Point /* Level 0 - Basic */
    {
        private BigInteger _x;
        private BigInteger _y;
    }
}
```

- Support for ___npcconfig class for imbedded npcc configuration parameters (e.g. NPC Level)
- Investigate support for partial classes in C#.NEO/C#.NPC

9. Summary

Sidebar: What is this a picture of? https://www.linkedin.com/groups/50758/50758-6372830537052155904

- It’s an architecture for a global NEO blockchain app
- ...the spoons represent Actors in the real world who in turn have personas (Business (and non-Business) Roles).
- The black phone and the red thing are apps (with smart contracts) - deployed on and consumers (Accessing) of the NEO blockchain (Application Software/Functions).
- The Lucite thing is an authenticated identity (Actor) for the Actor spoon on the far side of the table.
- The Sweet'n Low packets are his/her personas (Business/non-Business Roles).

9. Summary

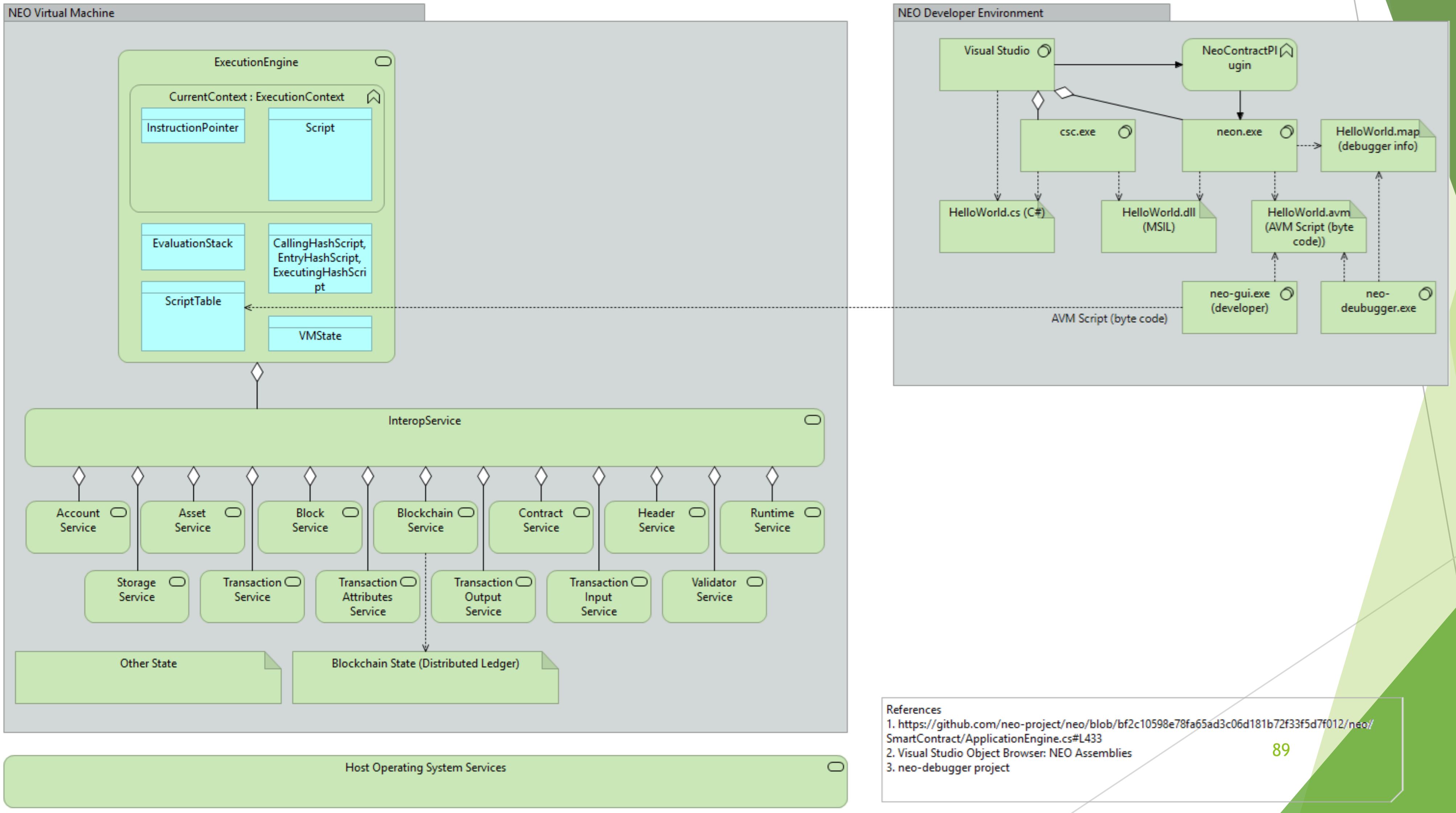
- NPC Level 0 Basic
- NPC Level 1 Managed
- NPC Level 2 Persistable
- NPC Level 3 Deletable (Bury/Tombstone)
- NPC Level 4 Collectible
- NPC Level 5 Extendible (Roadmap)
- NPC Level 6 Authorized (Roadmap)
- NPC Level 6 Optimized (Roadmap

Appendices

Appendix A - NEO Blockchain Architecture Reference Model (ARM)

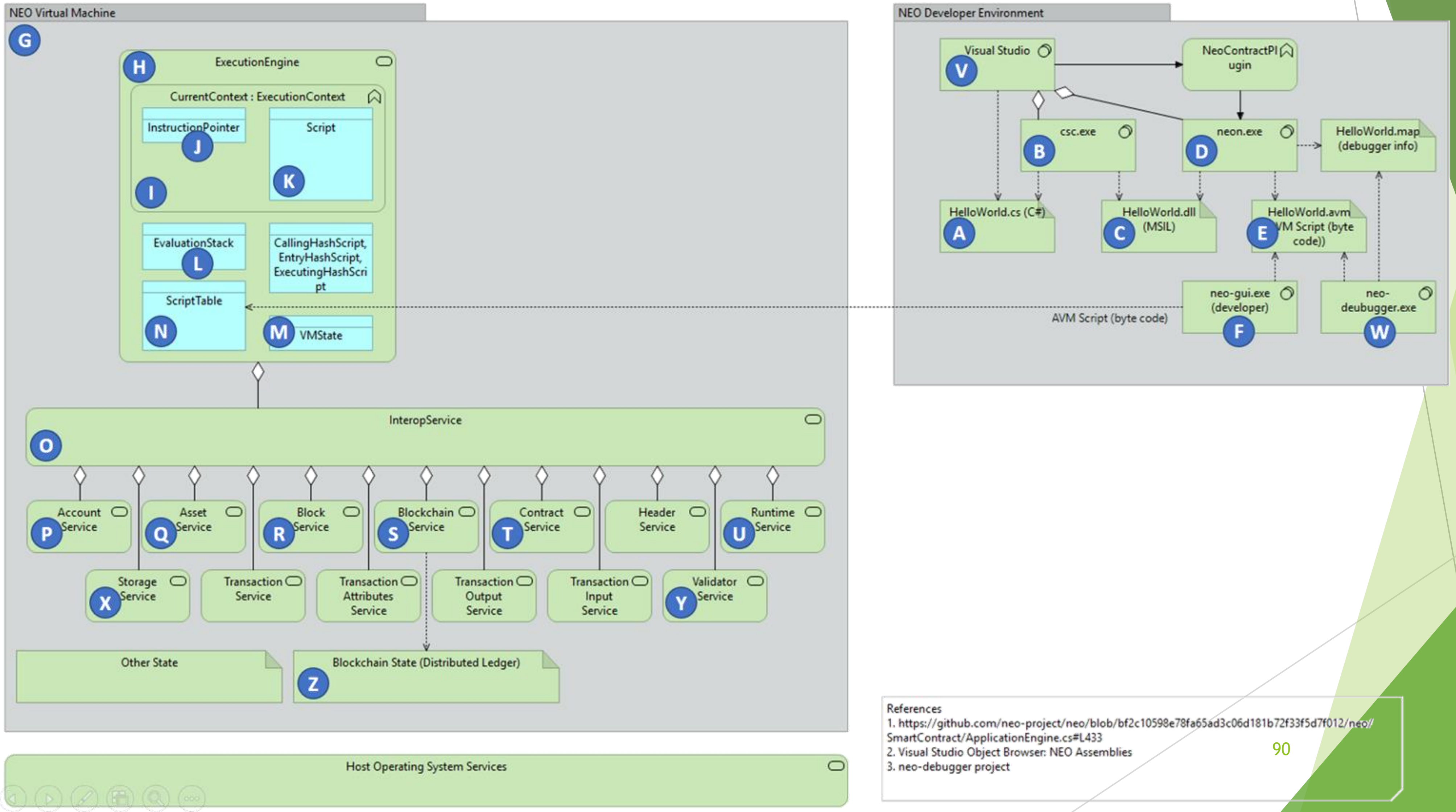
https://github.com/mwherman2000/neo-charm

NEO Blockchain Architecture Reference Model (ARM)



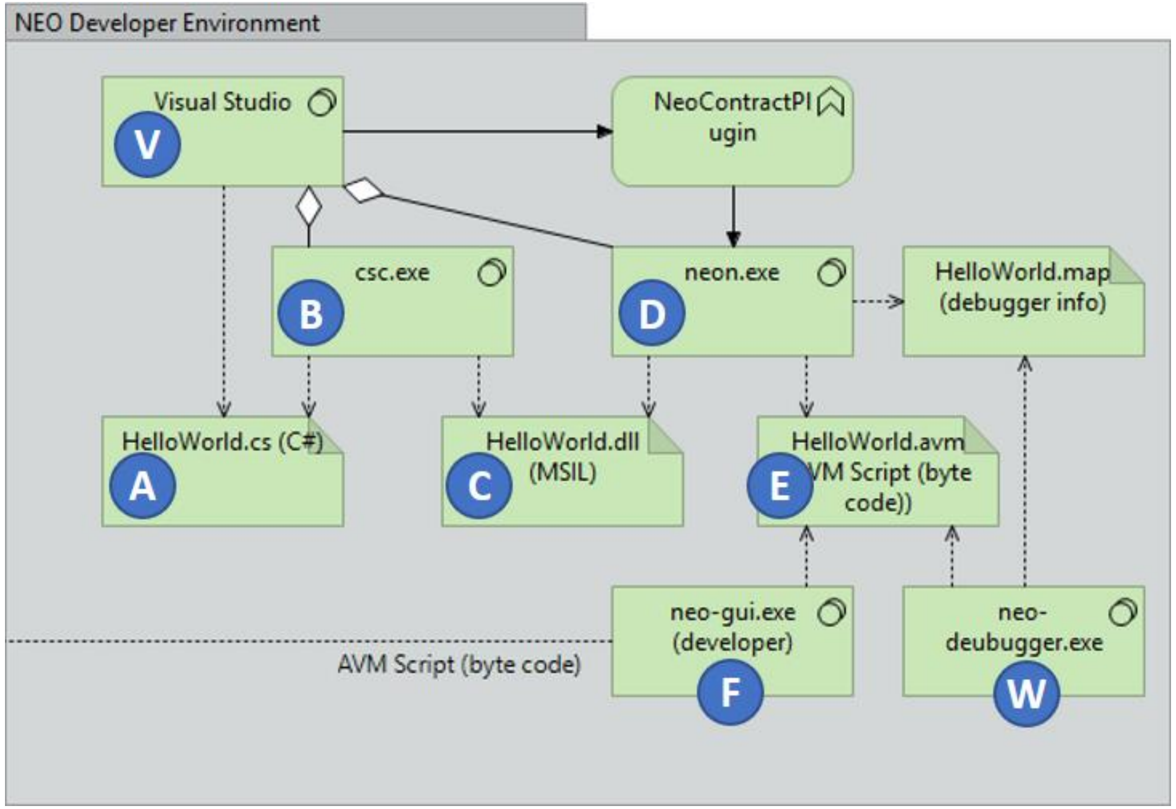
References
1. https://github.com/neo-project/neo/blob/bf2c10598e78fa65ad3c06d181b72f33f5d7f012/neo/SmartContract/ApplicationEngine.cs#L433
2. Visual Studio Object Browser: NEO Assemblies
3. neo-debugger project

NEO Blockchain ARM

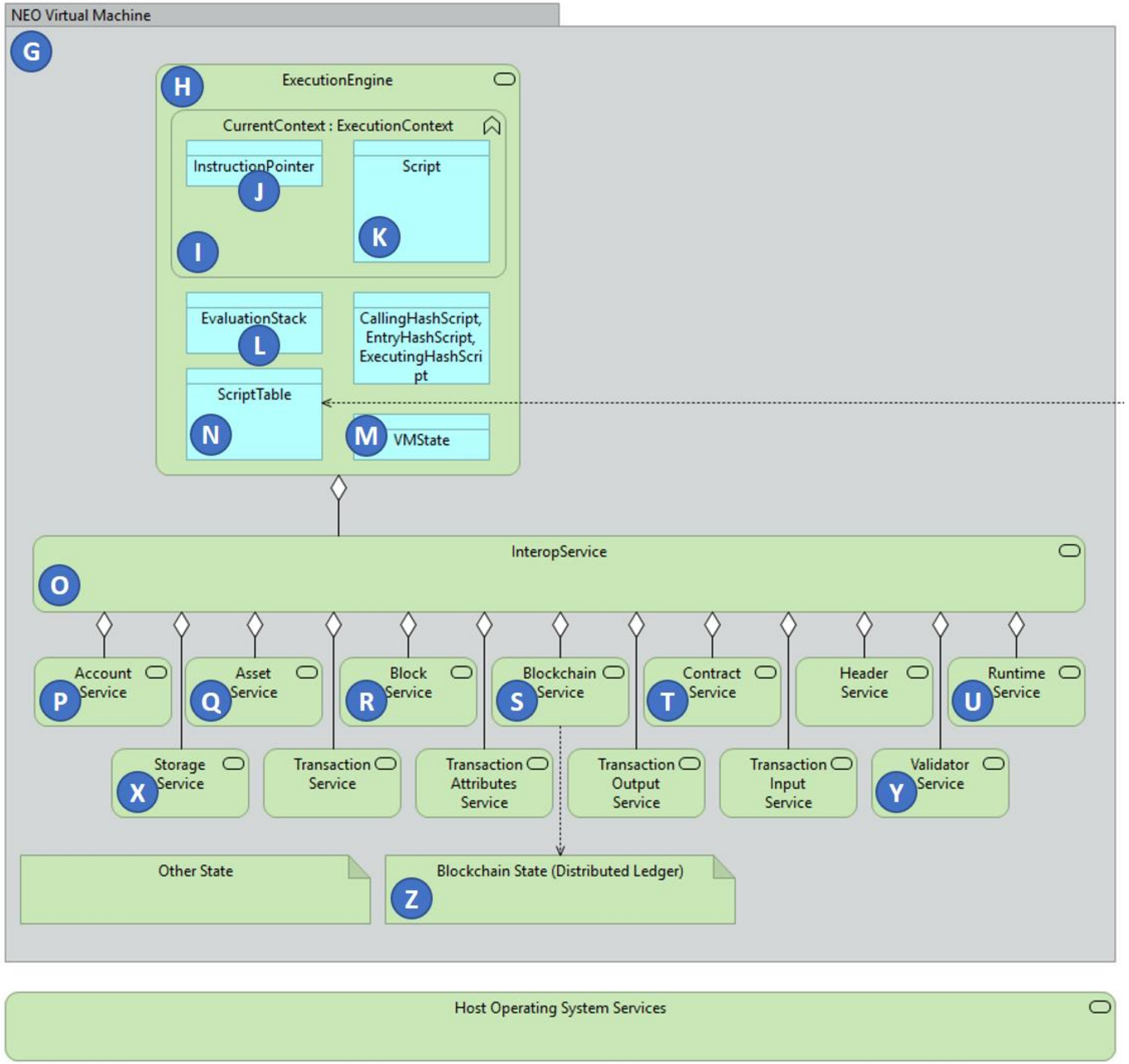


References
1. https://github.com/neo-project/neo/blob/bf2c10598e78fa65ad3c06d181b72f33f5d7f012/neo/SmartContract/ApplicationEngine.cs#L433
2. Visual Studio Object Browser: NEO Assemblies
3. neo-debugger project

NEO Developer Environment



NEO Virtual Machine



Questions?

Discord: Michael Herman (Toronto)

Email: mwherman@parallelspace.net

License and Copyright

MIT License

Copyright (c) 2018 Michael Herman (Toronto)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	ⓘ License and copyright notice
✓ Modification	✗ Warranty	
✓ Distribution		
✓ Private use		