Digging Deep Into LINQ
AKA: Fun With Expression Trees

dominion digital

process
technology
consulting
solutions

Justin Etheredge – C# MVP
http://www.CodeThinked.com
@JustinEtheredge

```
public delegate void MyDelegate(string val1,int val2);

public void Test()
{
    MyDelegate del = Method;
    del("val1", 2);
}

public void Method(string val1, int val2)
{
}
```

```
IEnumerable<Person> found = people.Where(p => p.FirstName == "Justin");
```

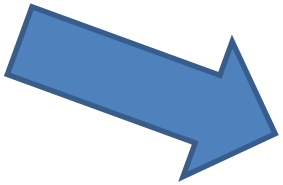
```
IEnumerable<Person> found = people.Where(person => person.FirstName == "Justin");
```

```
IEnumerable<Person> found = people.Where((person) => person.FirstName == "Justin");
```

```
IEnumerable<Person> found = people.Where((Person person) => person.FirstName == "Justin");
```

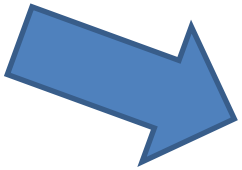
```
var found = people.Where((Person person) => { return person.FirstName == "Justin"; });
```

```
Action
```



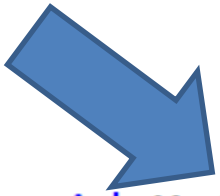
```
public void Method()
```

```
Action<string>
```



```
public void Method(string val1)
```

```
Action<string,int>
```



```
public void Method(string val1, int val2)
```

```
Func<Person>
```

```
public Person Method()
```

```
Func<Person,string>
```

```
public string Method(Person person)
```

```
Func<Person,string,string>
```

```
public string Method(Person person, string variable1)
```

LINQ
(Language Integrated Query)

```
var peop
    {
    };

IEnumera
```

```
e = "                    " }

eople                    stin");
```

LINQ to Objects

LINQ to SQL

LINQ to Entites

LINQ to XML

LINQ to DataSets

LINQ to ADO.NET Data Services

LINQ to Nhibernate

LINQ to CSV

```
var people = new[]
    {
        new Person { FirstName = "Justin", LastName = "Etheredge" }
    };

IEnumerable<Person> found = from p in people
                            where p.FirstName == "Justin"
                            select p;
```

```
IEnumerable<Person> found = CS$0$0000.Where<Person>(delegate (Person p) {
    return p.FirstName == "Justin";
});
```

```
IEnumerable<Person> found = people.Where(p => p.FirstName == "Justin");
```

```
IEnumerable<Person> found = from p in people
                            where p.FirstName == "Justin"
                            select p;
```

```
IEnumerable<Person> found = people.Where(p => p.FirstName == "Justin");
```

```
var people = new[]
    {
        new Person { FirstName = "Justin", LastName = "Etheredge" }
    };

IEnumerable<Person> found = people.Where(p => p.FirstName == "Justin");
```

```
p => p.FirstName == "Justin"
```

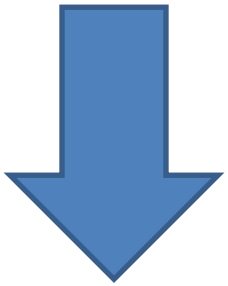
```
Func<Person, bool> del =
    delegate(Person person)
    {
        return person.FirstName == "Justin";
    };
```

```
public IEnumerable<Person> Where(IEnumerable<Person> people,
                                 Func<Person, bool> filter)
{
    foreach (Person person in people)
    {
        if (filter(person))
        {
            yield return person;
        }
    }
}
```

LINQ to SQL

Entity Framework

```
IEnumerable<Person> found = from p in people
                            where p.FirstName == "Justin"
                            select p;
```

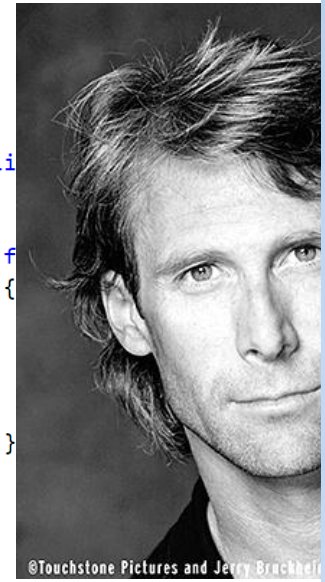


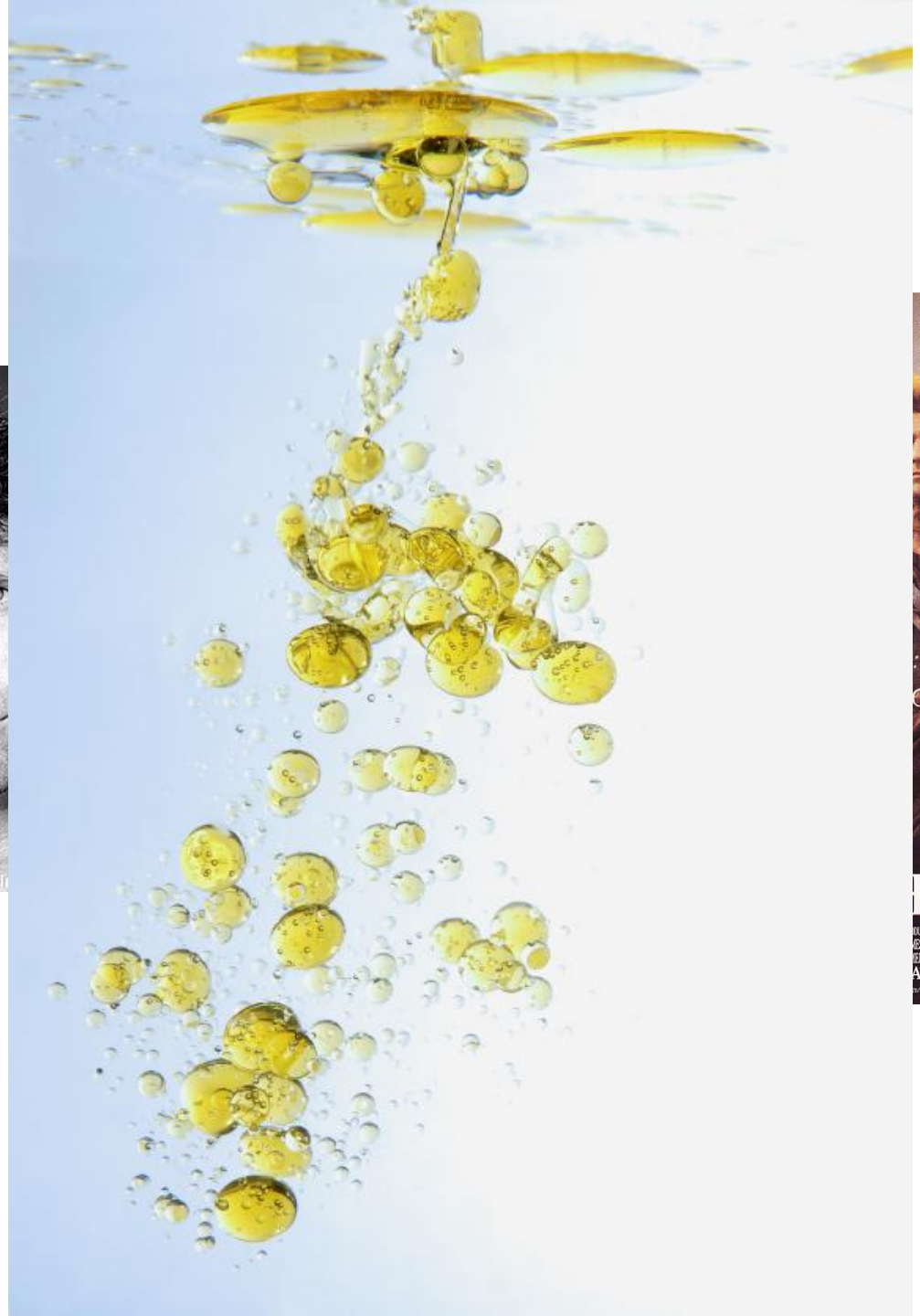
SELECT Id, FirstName, LastName FROM People WHERE FirstName = “Justin”

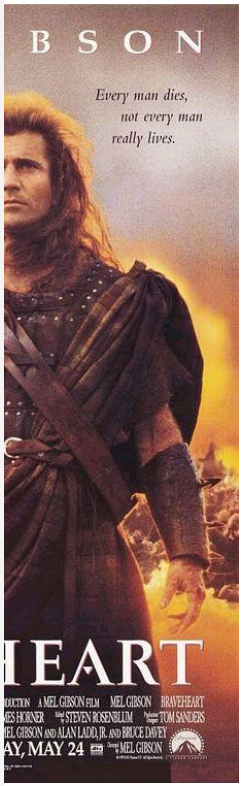
```
publi

{
    f
    {

    }
}
```


©Touchstone Pictures and Jerry Bruckheim



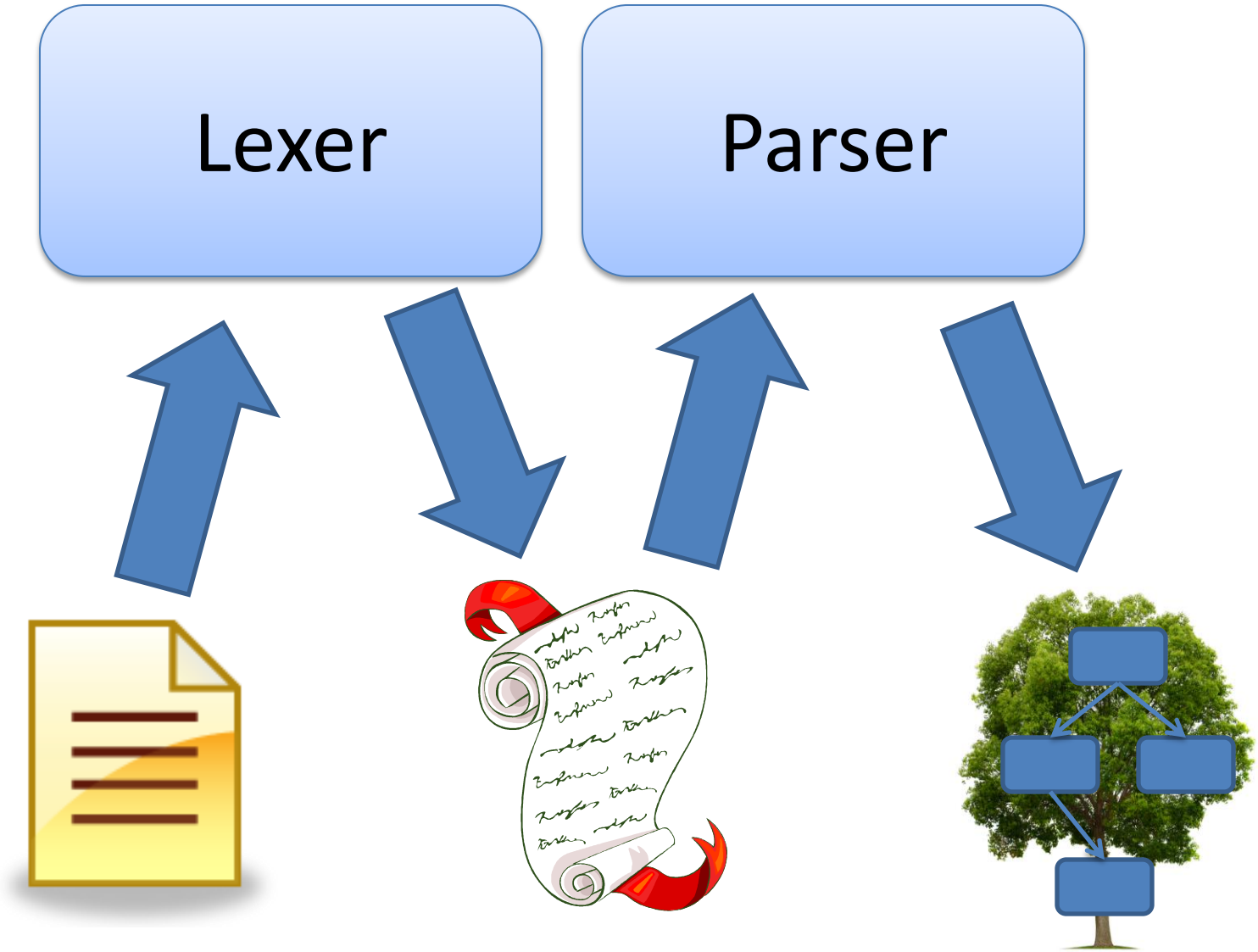

B S O N
Every man dies,
not every man
really lives.
HEART
A MEL GIBSON FILM
MEL GIBSON
BRAVEHEART
STEVEN ROSENBLUM
TOM SANDERS
MEL GIBSON AND ALAN LADD, JR. AND BRUCE DAVEY
AY, MAY 24
MEL GIBSON

```
.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    .maxstack 4
    .locals init (
        [0] class ExpressionTreeTest.Person[] people,
        [1] class [mscorlib]System.Collections.Generic.IEnumerable`1<class ExpressionTreeTest.Person> found,
        [2] class ExpressionTreeTest.Person <>g__initLocal0,
        [3] class ExpressionTreeTest.Person[] CS$0$0000)
    L_0000: nop
    L_0001: ldc.i4.1
    L_0002: newarr ExpressionTreeTest.Person
    L_0007: stloc.3
    L_0008: ldloc.3
    L_0009: ldc.i4.0
    L_000a: newobj instance void ExpressionTreeTest.Person::.ctor()
    L_000f: stloc.2
    L_0010: ldloc.2
    L_0011: ldstr "Justin"
    L_0016: callvirt instance void ExpressionTreeTest.Person::set_FirstName(string)
    L_001b: nop
    L_001c: ldloc.2
    L_001d: ldstr "Etheredge"
    L_0022: callvirt instance void ExpressionTreeTest.Person::set_LastName(string)
    L_0027: nop
    L_0028: ldloc.2
    L_0029: stelem.ref
    L_002a: ldloc.3
    L_002b: stloc.0
    L_002c: ldloc.0
    L_002d: ldsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_0032: brtrue.s L_0047
    L_0034: ldnull
    L_0035: ldftn bool ExpressionTreeTest.Program::<Main>b__1(class ExpressionTreeTest.Person)
    L_003b: newobj instance void [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool>::.ctor(object, native int)
    L_0040: stsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_0045: br.s L_0047
    L_0047: ldsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_004c: call class [mscorlib]System.Collections.Generic.IEnumerable`1<!!0> [System.Core]System.Linq.Enumerable::Where<class ExpressionTreeTest.Person>(class [m
    L_0051: stloc.1
    L_0052: ret
}
```

?

Expression Trees!

Expression Tree?

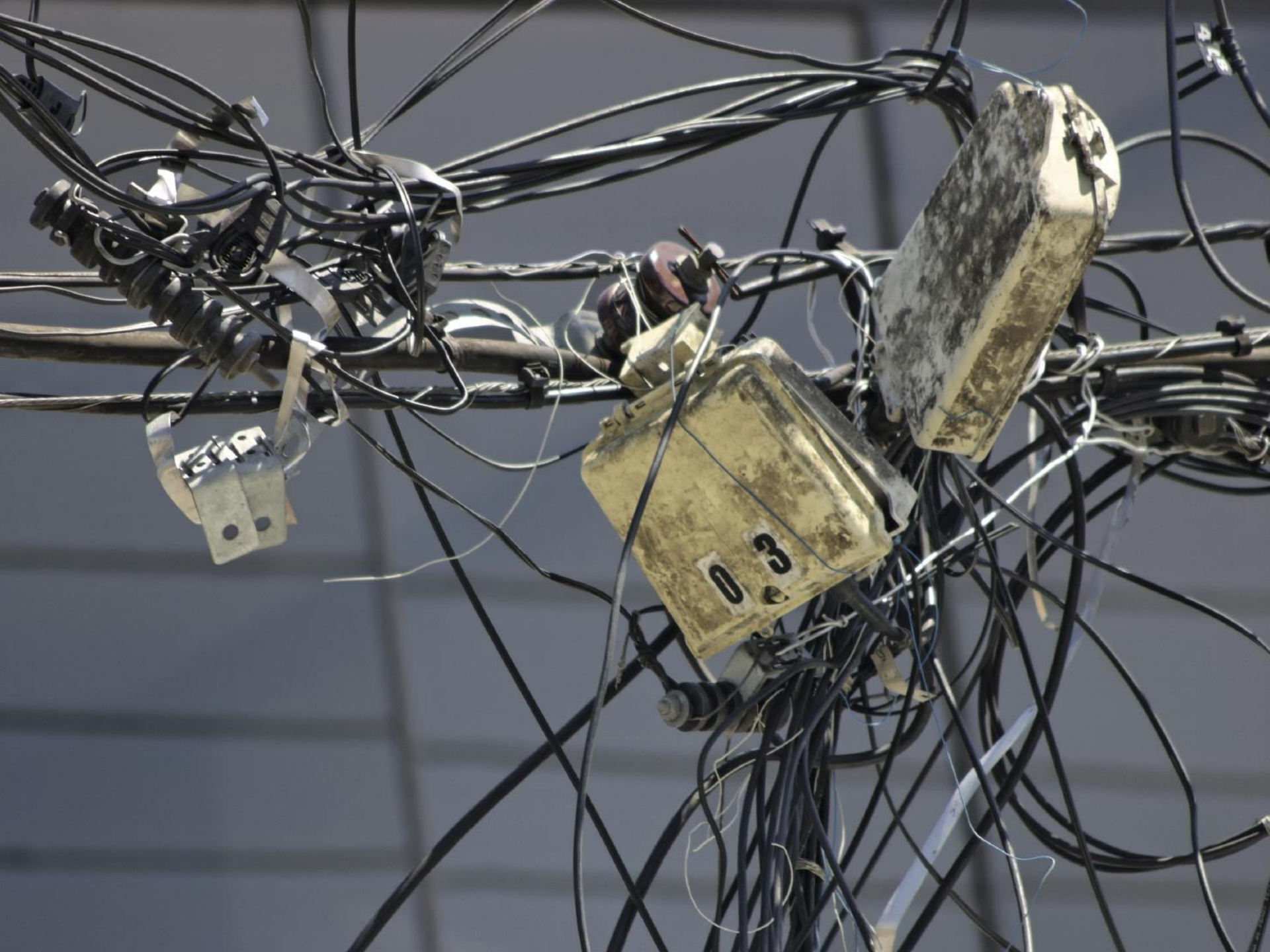

Lexer
Parser

Abstract Syntax Tree

Expression Tree

System.Linq.Expressions

Demo: Manually Building an Expression Tree

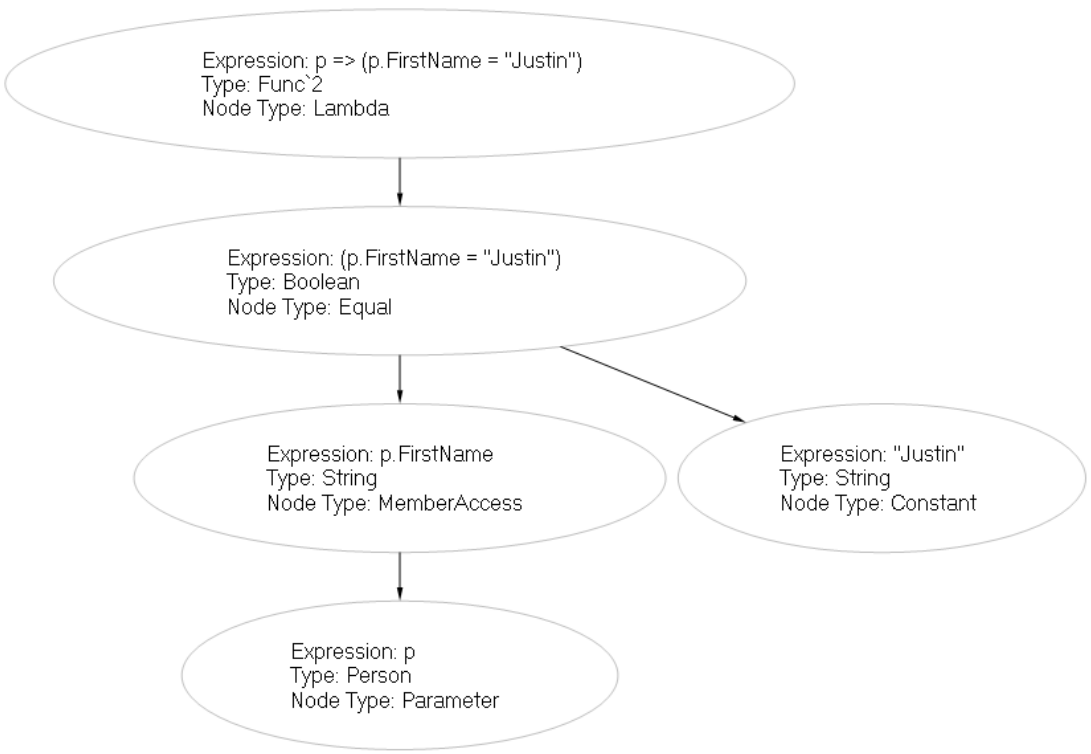

03

Expression<TDelegate>

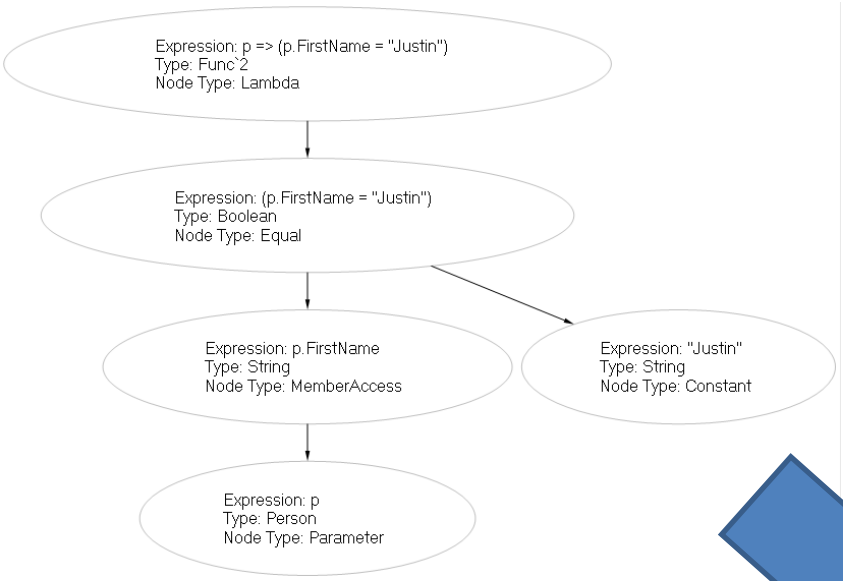
Demo: Easily Constructing Expression Trees

```
.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    .maxstack 4
    .locals init (
        [0] class ExpressionTreeTest.Person[] people,
        [1] class [mscorlib]System.Collections.Generic.IEnumerable`1<class ExpressionTreeTest.Person> found,
        [2] class ExpressionTreeTest.Person <>g__initLocal0,
        [3] class ExpressionTreeTest.Person[] CS$0$0000)
    L_0000: nop
    L_0001: ldc.i4.1
    L_0002: newarr ExpressionTreeTest.Person
    L_0007: stloc.3
    L_0008: ldloc.3
    L_0009: ldc.i4.0
    L_000a: newobj instance void ExpressionTreeTest.Person::.ctor()
    L_000f: stloc.2
    L_0010: ldloc.2
    L_0011: ldstr "Justin"
    L_0016: callvirt instance void ExpressionTreeTest.Person::set_FirstName(string)
    L_001b: nop
    L_001c: ldloc.2
    L_001d: ldstr "Etheredge"
    L_0022: callvirt instance void ExpressionTreeTest.Person::set_LastName(string)
    L_0027: nop
    L_0028: ldloc.2
    L_0029: stelem.ref
    L_002a: ldloc.3
    L_002b: stloc.0
    L_002c: ldloc.0
    L_002d: ldsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_0032: brtrue.s L_0047
    L_0034: ldnull
    L_0035: ldftn bool ExpressionTreeTest.Program::<Main>b__1(class ExpressionTreeTest.Person)
    L_003b: newobj instance void [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool>::.ctor(object, native int)
    L_0040: stsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_0045: br.s L_0047
    L_0047: ldsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_004c: call class [mscorlib]System.Collections.Generic.IEnumerable`1<!!0> [System.Core]System.Linq.Enumerable::Where<class ExpressionTreeTest.Person>(class [m
    L_0051: stloc.1
    L_0052: ret
}
```

SELECT Id, FirstName, LastName FROM People WHERE FirstName = “Justin”



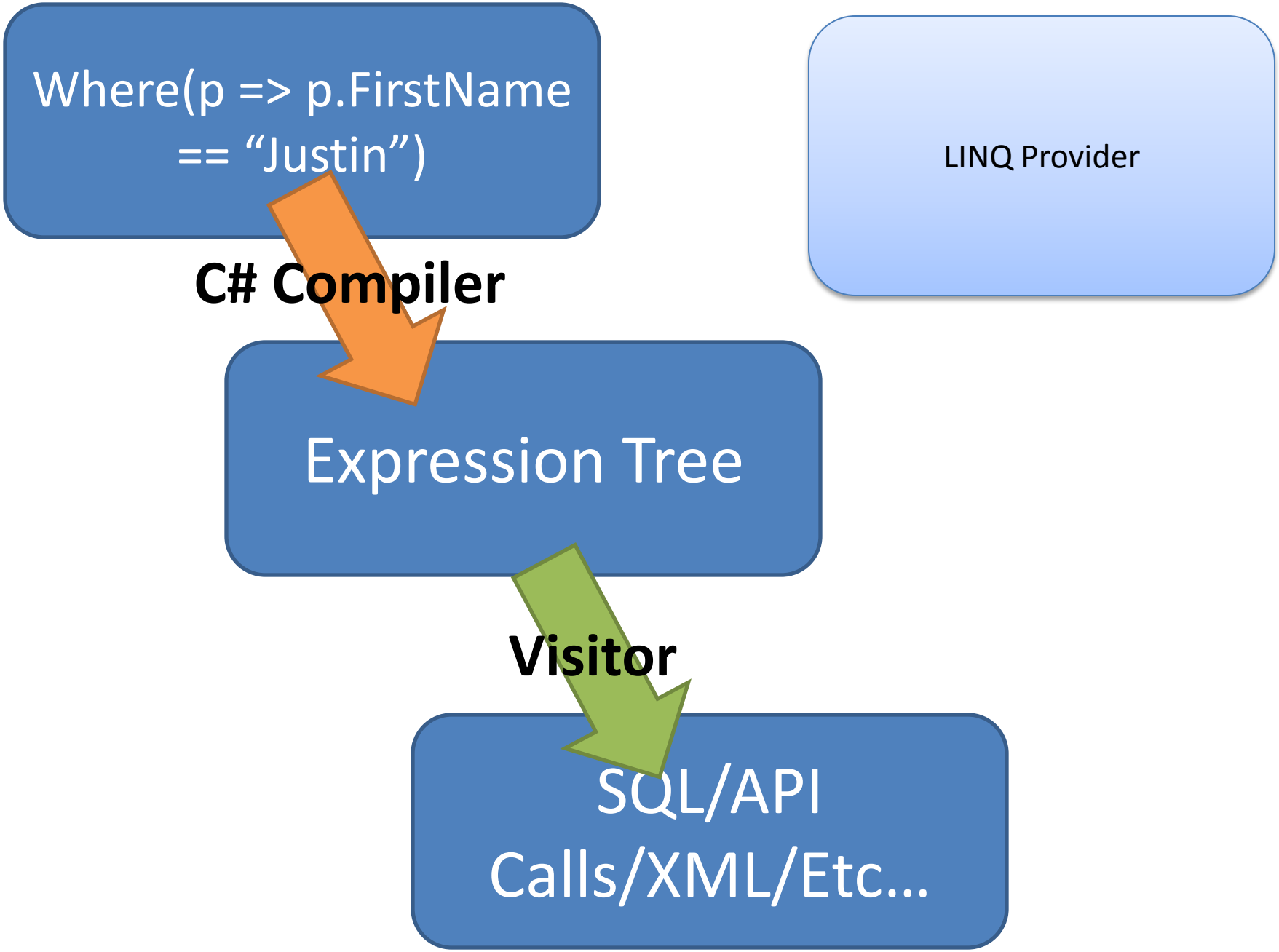
SELECT Id, FirstName, LastName FROM People WHERE FirstName = “Justin”




```
.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    .maxstack 4
    .locals init (
        [0] class ExpressionTreeTest.Person[] people,
        [1] class [mscorlib]System.Collections.Generic.IEnumerable`1<class ExpressionTreeTest.Person> found,
        [2] class ExpressionTreeTest.Person <>g__initLocal0,
        [3] class ExpressionTreeTest.Person[] CS$0$0000)
    L_0000: nop
    L_0001: ldc.i4.1
    L_0002: newarr ExpressionTreeTest.Person
    L_0007: stloc.3
    L_0008: ldloc.3
    L_0009: ldc.i4.0
    L_000a: newobj instance void ExpressionTreeTest.Person::.ctor()
    L_000f: stloc.2
    L_0010: ldloc.2
    L_0011: ldstr "Justin"
    L_0016: callvirt instance void ExpressionTreeTest.Person::set_FirstName(string)
    L_001b: nop
    L_001c: ldloc.2
    L_001d: ldstr "Etheredge"
    L_0022: callvirt instance void ExpressionTreeTest.Person::set_LastName(string)
    L_0027: nop
    L_0028: ldloc.2
    L_0029: stelem.ref
    L_002a: ldloc.3
    L_002b: stloc.0
    L_002c: ldloc.0
    L_002d: ldsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_0032: brtrue.s L_0047
    L_0034: ldnull
    L_0035: ldftn bool ExpressionTreeTest.Program::<Main>b__1(class ExpressionTreeTest.Person)
    L_003b: newobj instance void [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool>::.ctor(object, native int)
    L_0040: stsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_0045: br.s L_0047
    L_0047: ldsfld class [mscorlib]System.Func`2<class ExpressionTreeTest.Person, bool> ExpressionTreeTest.Program::CS$<>9__CachedAnonymousMethodDelegate2
    L_004c: call class [mscorlib]System.Collections.Generic.IEnumerable`1<!!0> [System.Core]System.Linq.Enumerable::Where<class ExpressionTreeTest.Person>(class [m
    L_0051: stloc.1
    L_0052: ret
}
```

System.Linq.Expressions.ExpressionVisitor

Demo: Walking The Expression Tree


Where(p => p.FirstName == “Justin”)
LINQ Provider
C# Compiler
Expression Tree
Visitor
SQL/API Calls/XML/Etc…

LINQExtender

http://linqextender.codeplex.com/

Demo:
LINQ To

Demo: Getting Runtime Info

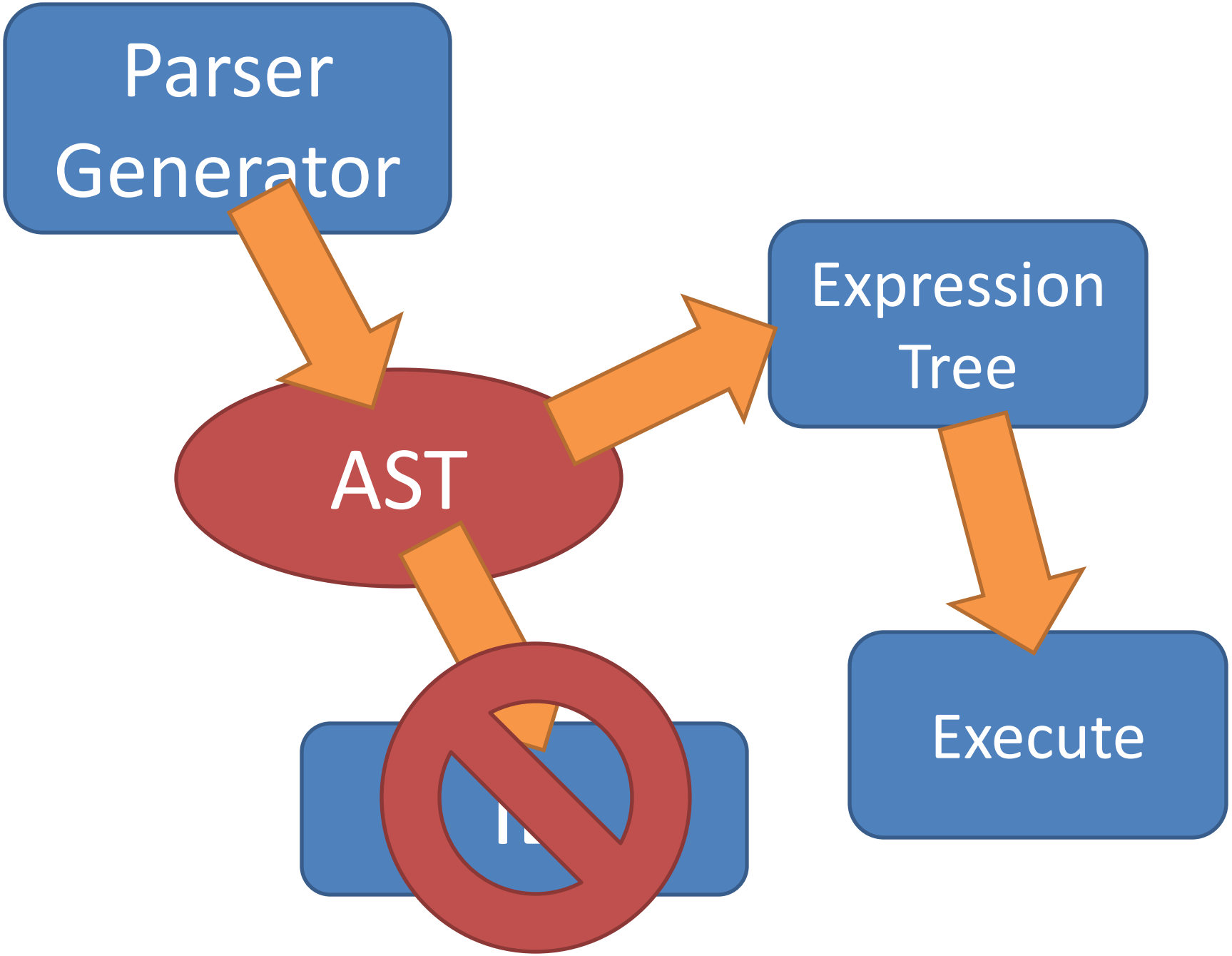
Demo:

Demo: Faster Than Reflection

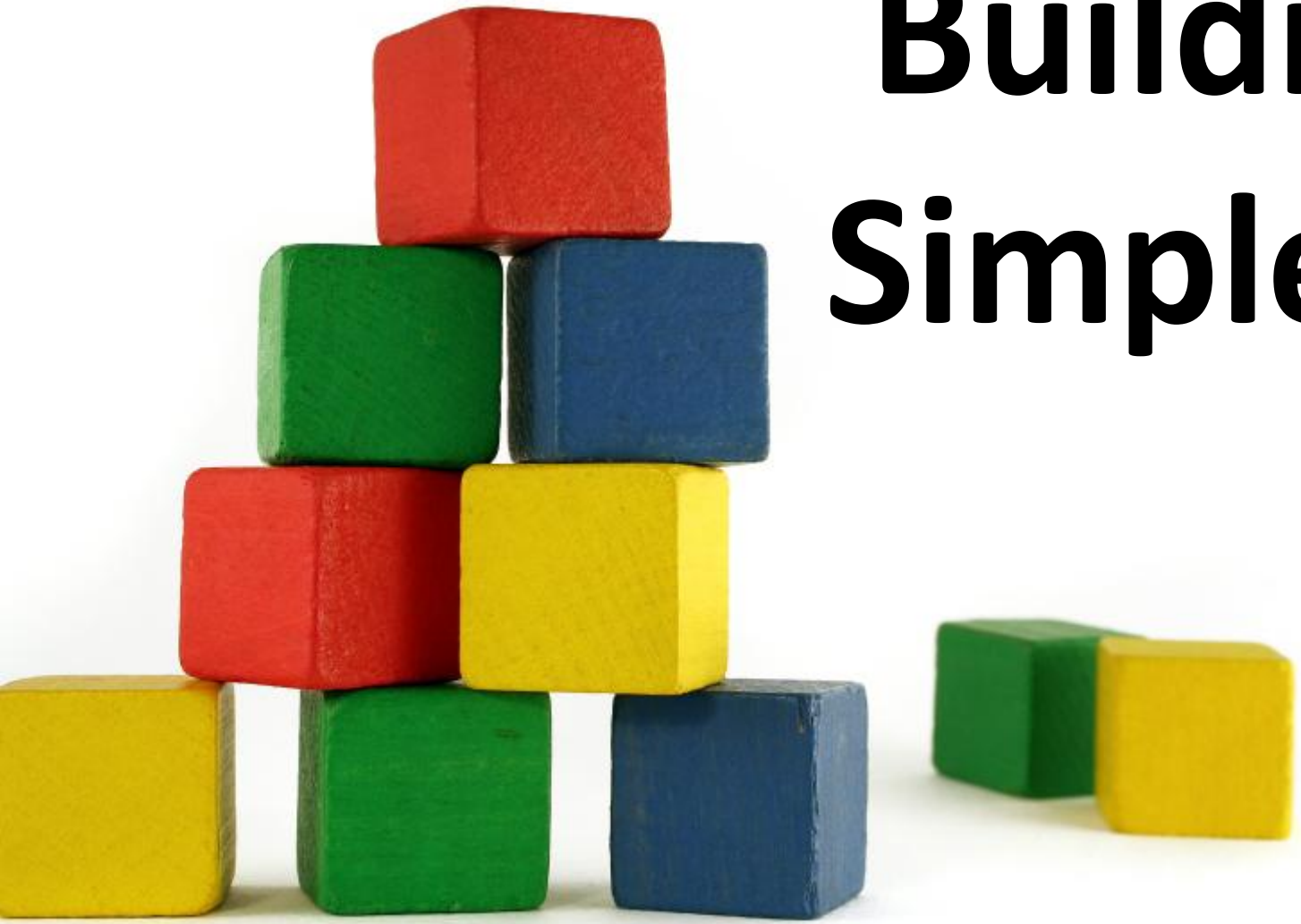
Nate Kohari
Kohari.org

Microsoft®
Dynamic Language Runtime™

Parser Generator


Parser Generator
AST
Expression Tree
Execute

Demo: Building a Simple DSL



Questions?

Thanks!

Justin Etheredge
http://www.codethinked.com
justin@etheredge.us