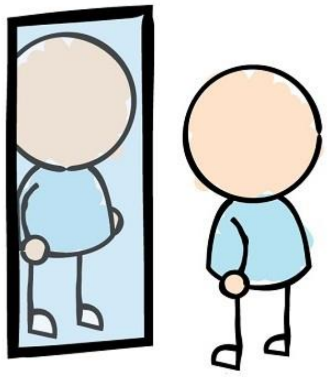
REFLECTION

MODULE 7

November 2015

What is Reflection

Definition - to look back to itself



Run Time Type Information

Discovering class information solely at run time

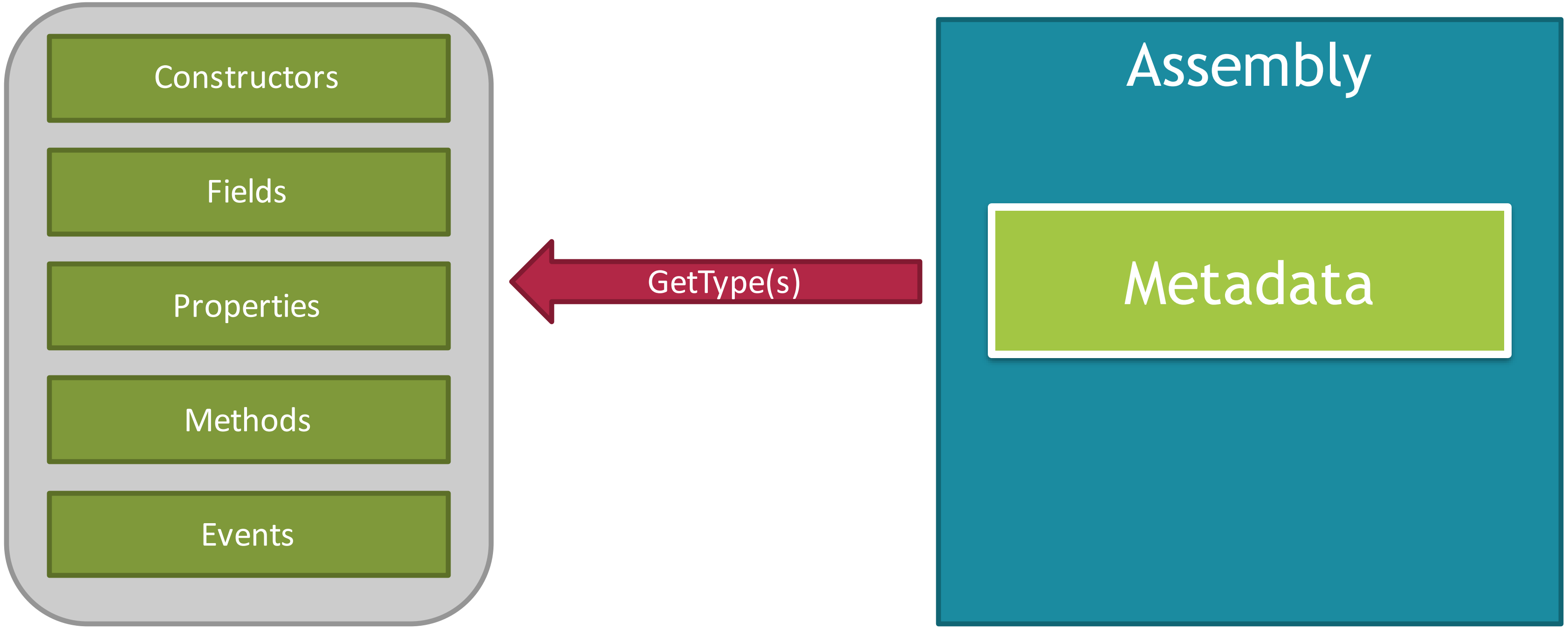
.Net Reflection

Assembly Access based on metadata

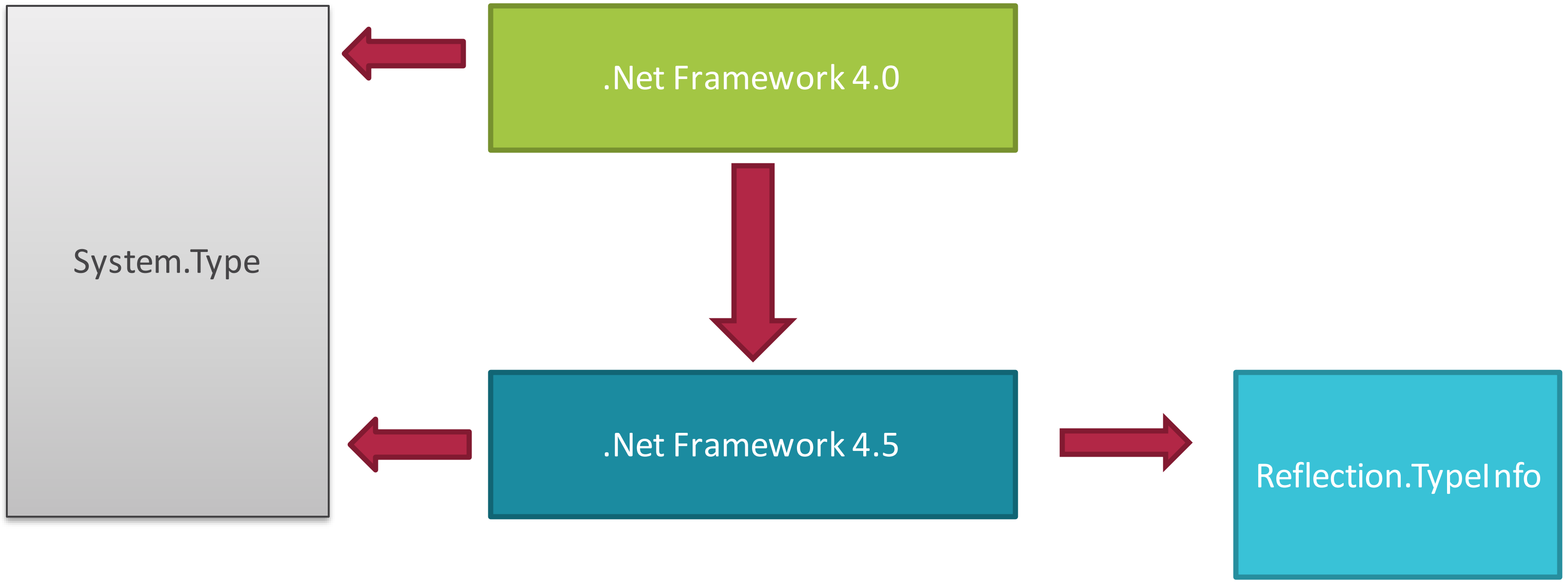
System.Type

- Base of all reflection classes and properties - primary way to access metadata
  - Represents all kinds of types (class, interface, enumeration…)
- GetType() is a Method of System.Object
- typeof() operator returns Type object
- A Type is unique per type
- Only one instance of Type per type per application domain

System.Type


Constructors
Fields
Properties
Methods
Events
GetType(s)
Assembly
Metadata

TypeInfo


System.Type
.Net Framework 4.0
.Net Framework 4.5
Reflection.TypeInfo

MemberInfo

- Provides access to member metadata.
- MemberType propery is useful to determine the type of a member
- Ancestor to all other ???Info classes
- Actually an ancestor to Type

???Info Classes

- PropertyInfo
  - GetGetMethod; GetSetMethod
- FieldInfo
  - GetValue; SetValue
- MethodInfo
  - Invoke
- ConstructorInfo
- EventInfo

System.Activator

- A generalized way for creating objects at runtime
- CreateInstance method - generic and regular
- Other Uses

Reflection and Generics

- Difference between regular types, generic types, and specialized types
- Type.IsGenericType
- Type.IsGenericTypeDefinition
- Type.MakeGenericType
- MethodInfo.IsGenericMethod
- MethodInfo.IsGenericMethodDefinition

Reflection.Emit

- Creating, modifying and running of code in runtime
- Does not use C#, uses MSIL
- Very complex, very powerfull

Performance

- As expected, direct access is much faster than reflection
- Performance hit – anywhere from 5 to 150 times slower
- Avoid using lots of reflection in tight loops
- We should weight if the usefulness in the scenario justifies the slowdown.

Questions