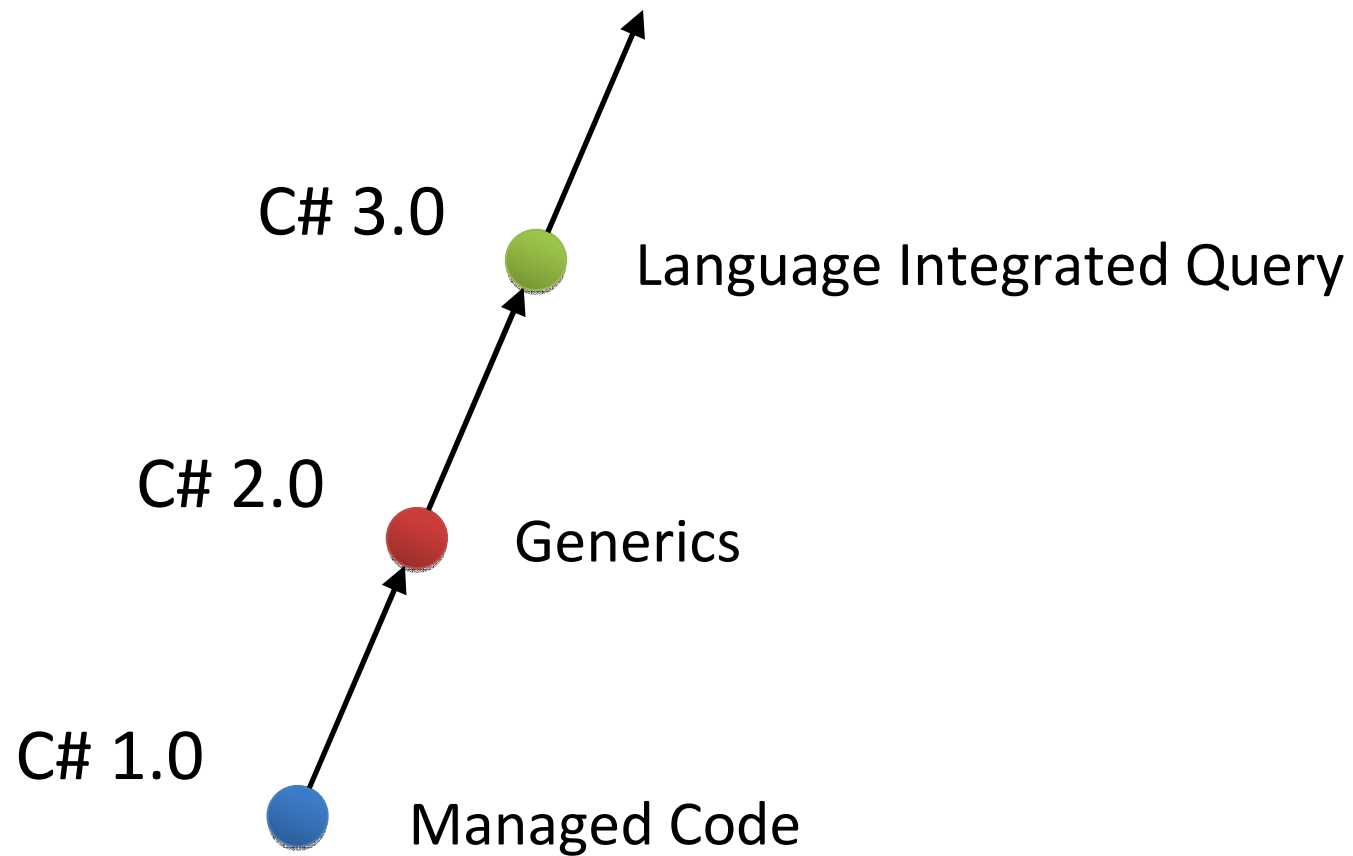# C# 3.0 and LINQ

Anders Hejlsberg
Microsoft Technical Fellow
andersh@microsoft.com

The Evolution of C#

C# 3.0

Language Integrated Query

C# 2.0

Generics

C# 1.0

Managed Code

C# 3.0 Design Goals

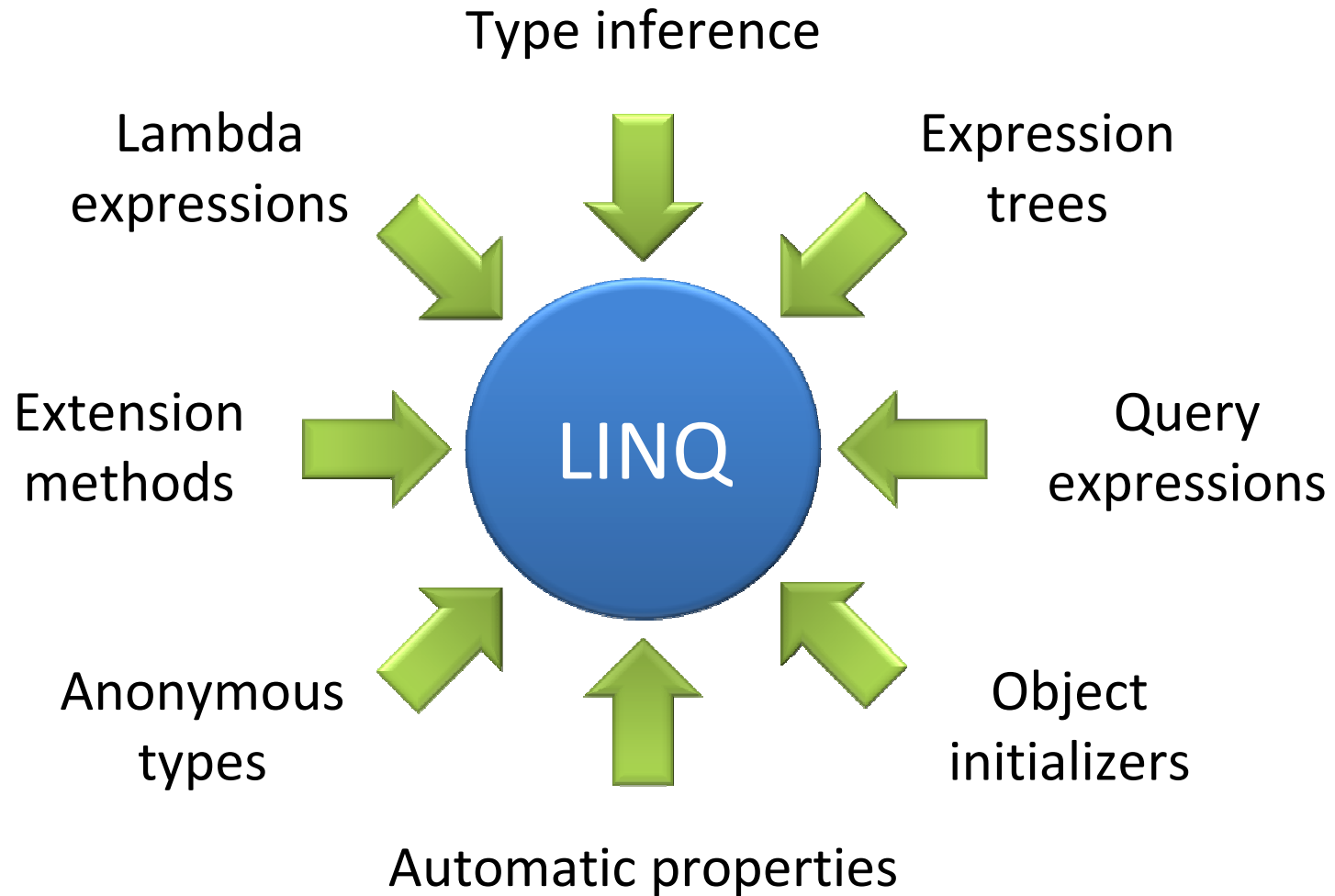**Integrate objects, relational data, and XML**

Increase conciseness of language

Add functional programming constructs

Isolate language from specific APIs

Remain 100% backwards compatible

C# 3.0 Features

Type inference

Lambda expressions

Expression trees

Extension methods

LINQ

Query expressions

Anonymous types

Object initializers

Automatic properties

Deferred Query Execution

```
Customer[] custs = SampleData.GetCustomers();

var query = custs.Where(c => c.City == "London").Select(c => c.Name);

string[] names = query.ToArray();
```

custs

ID	Name	Phone

names

Where

Select

c => c.City == "London"

c => c.Name

Query Expressions

Starts with **from**

Zero or more **from**, **join**, **let**, **where**, or **orderby**

Ends with **select** or **group by**

Optional **into** continuation

from *id* **in** *source*
{ **from** *id* **in** *source* |
  join *id* **in** *source* **on** *expr* **equals** *expr* [**into** *id*] |
  let *id* = *expr* |
  where *condition* |
  orderby *ordering, ordering, …* }
  select *expr* | **group** *expr* **by** *key*
[**into** *id query*]

Microsoft®

Your potential. Our passion.™