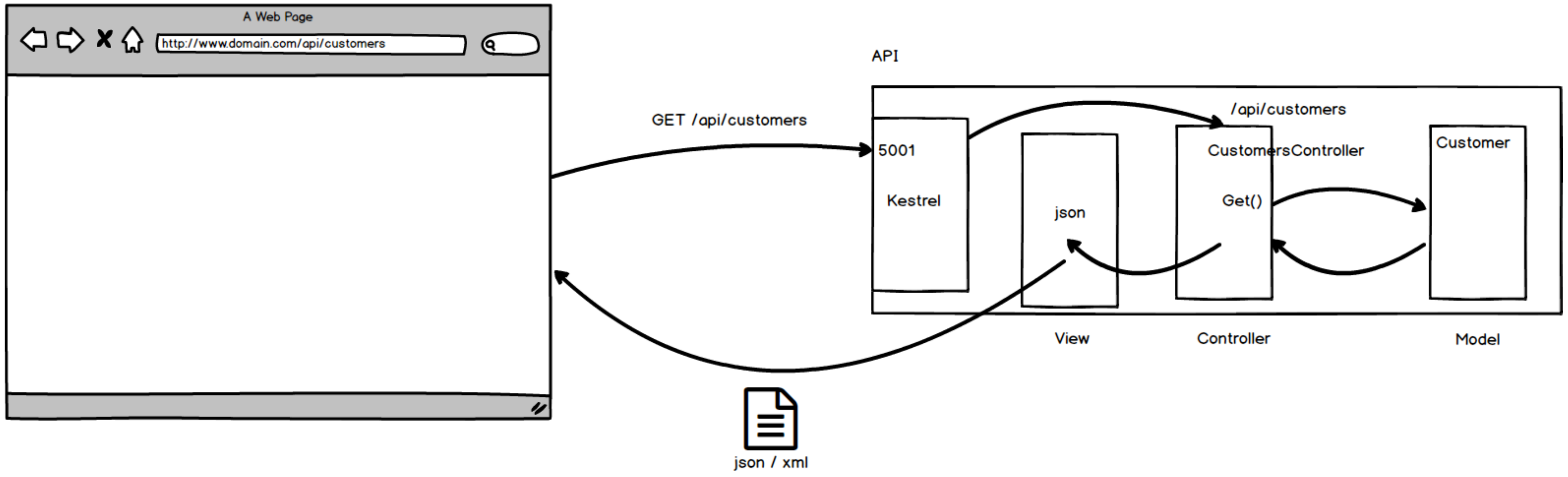
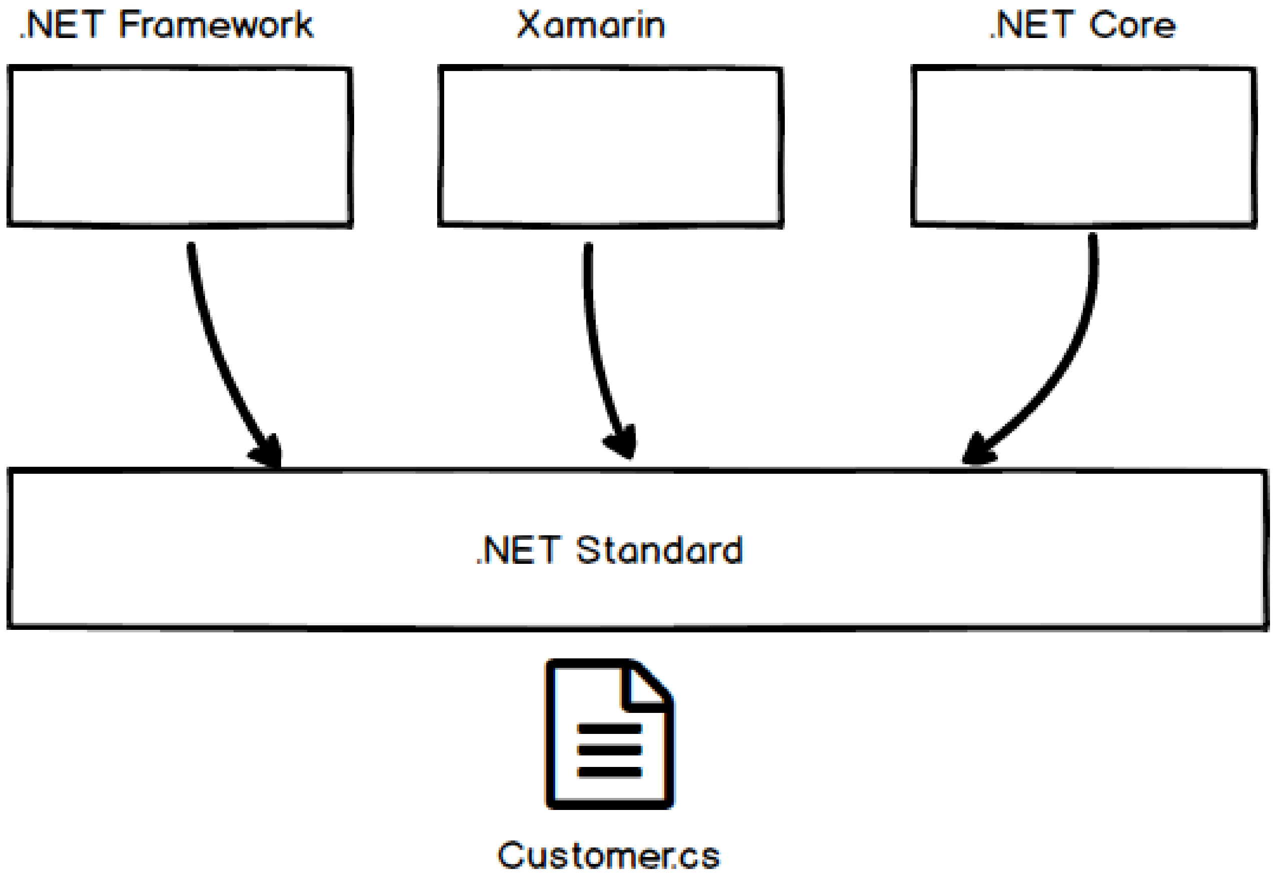

A Web Page
http://www.domain.com/api/customers
GET /api/customers
API
/api/customers
5001
Kestrel
json
CustomersController
Get()
Customer
View
Controller
Model
json / xml



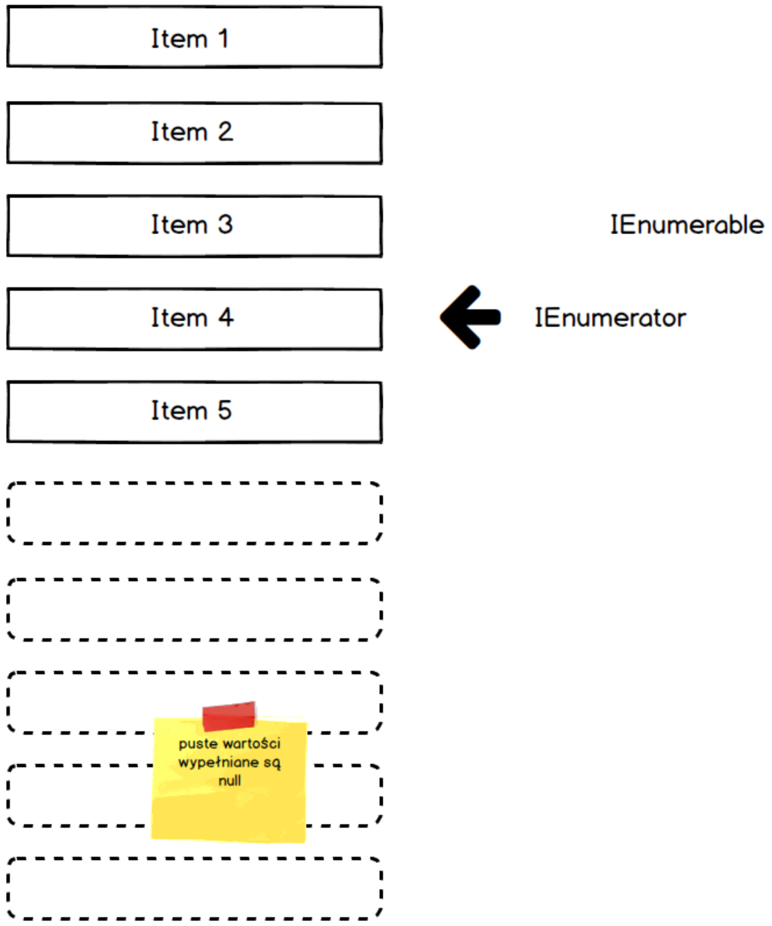
Projekty zgodne z .NET Standard mogą być wskazywane (Add reference) przez dowolny inny framework

Tablica (Array)

```
byte[] numbers = new byte[100];

numbers[0] = 100
numbers[1] = 50
numbers[2] = 56
numbers[3] = 99
```

stała liczba elementów określonego typu

Po utworzeniu tablicy nie można zmienić jej rozmiaru.



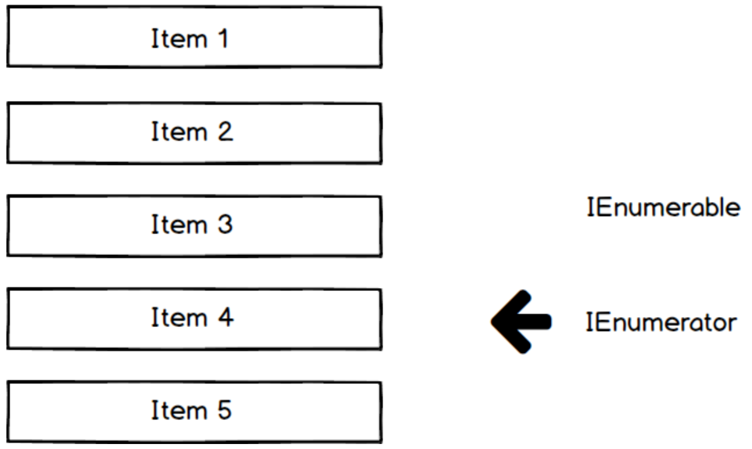
```
foreach(byte number in numbers) { }
```

Kolekcja (Collection)

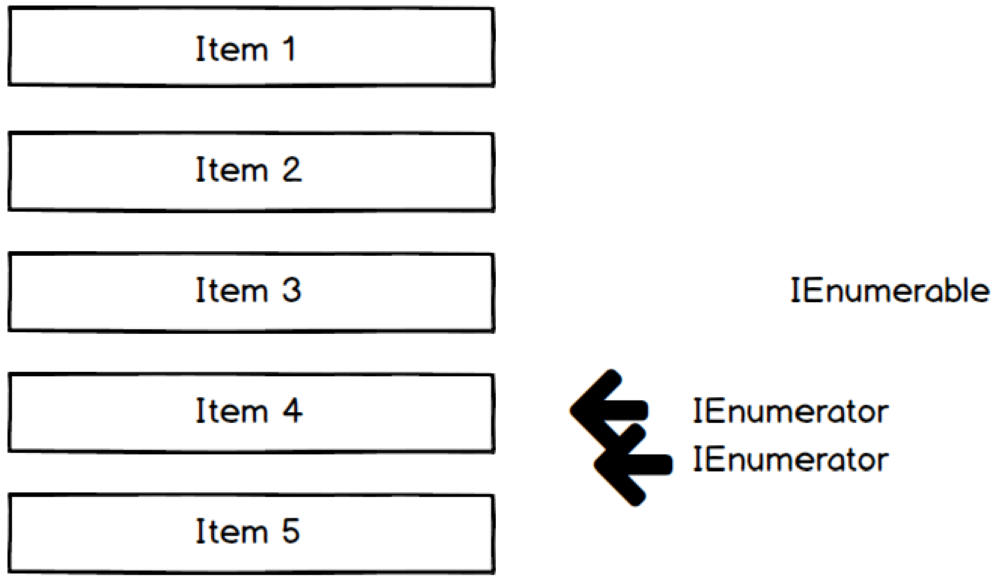
```
ICollection<Customer> customers = new List<Customer>();
customers.Add(new Customer());
customers.Add(new Customer());
customers.Add(new Customer());
customers.Add(new Customer());
customers.Add(new Customer());

foreach(Customer customer in customers) { }
```

zmienna liczba elementów określonego typu

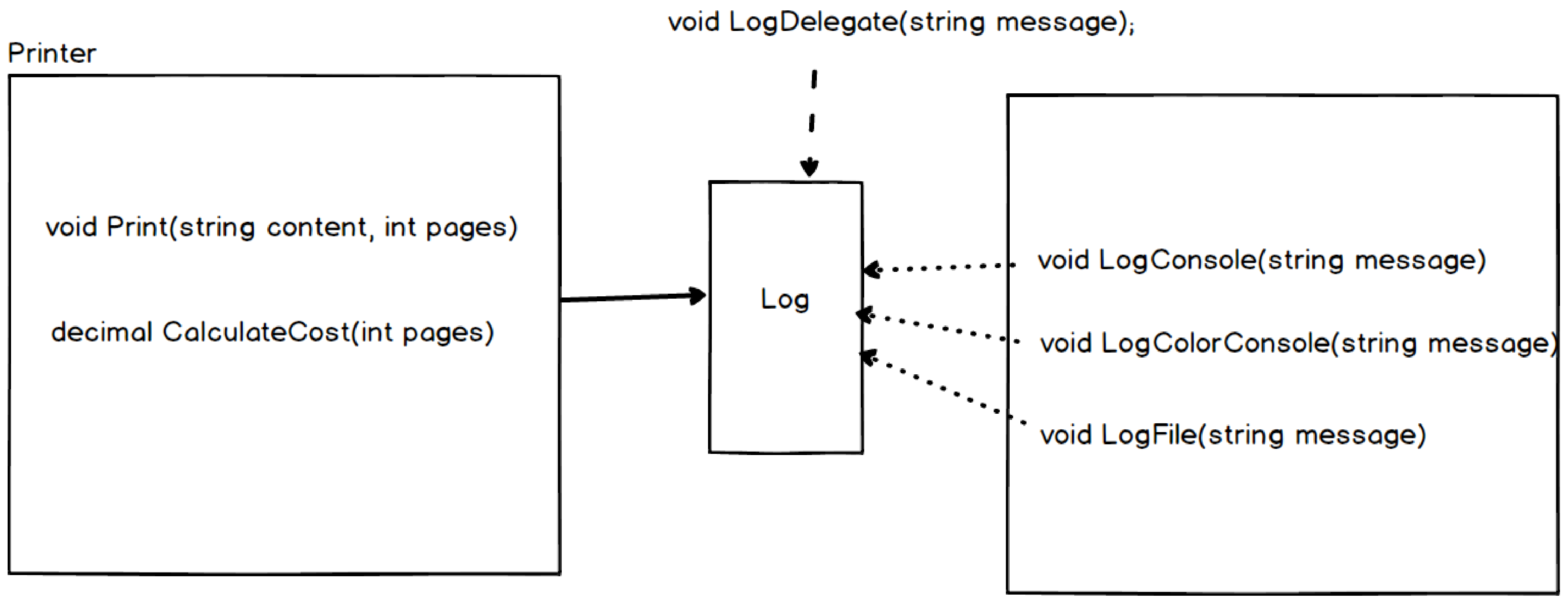
Po utworzeniu kolekcji jej rozmiar może ulec zmianie przy dodawaniu lub usuwaniu elementów



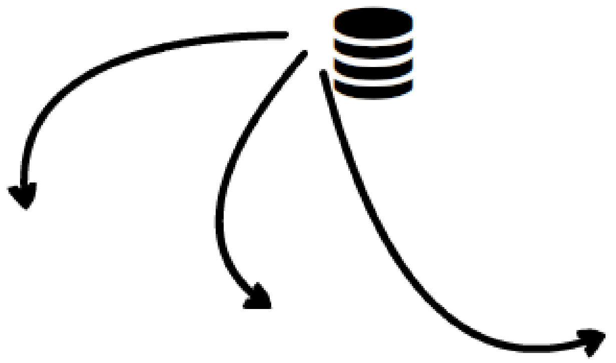
Tablica (Array)



Delegates


void LogDelegate(string message);
Printer
void Print(string content, int pages)
decimal CalculateCost(int pages)
Log
void LogConsole(string message)
void LogColorConsole(string message)
void LogFile(string message)

Vehicle1	□	■	□	■
Vehicle2	■	■	□	■
Vehicle3	■	□	□	□



Dictionary<Vehicle, Delegate>

bool IsOpenDoor(byte[] flags]

```
Vehicle1    {  if    }
Vehicle2    {  if    }
Vehicle3    {  if    }
```

imperatywny

```
public int Add(int x, int y)
{
    return x + y;
}
```

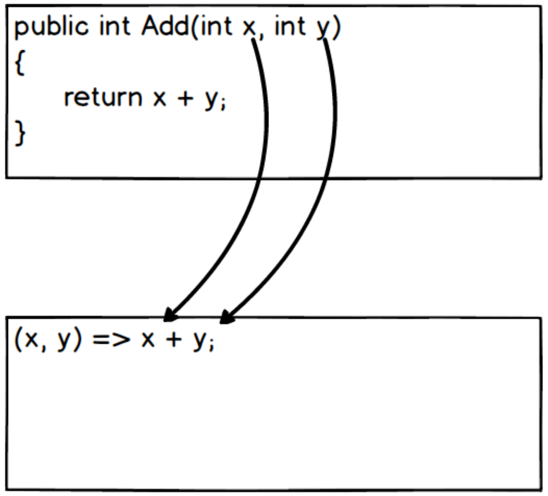
```
f(x, y) = x + y  x,y nalezy do int
```

```
g(x, y) = x + y  x,y nalezy do int
```

```
\
/ \ (x, y) -> x + y  x,y nalezy do int
```

```
public bool Filter(Product product)
{
    return product.UnitPrice > 100;
}
```

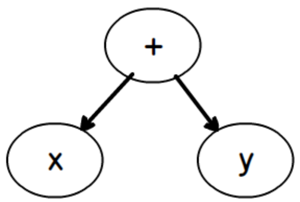
```
delegate (Product product)
{
    return product.UnitPrice > 100;
}
```



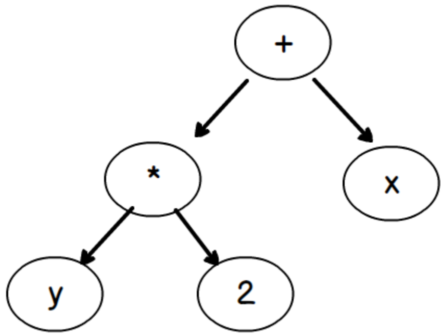
```
public int Add(int x, int y)
{
    return x + y;
}
```

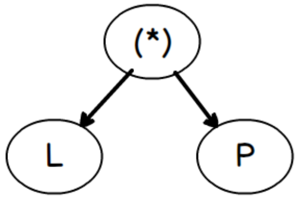
```
(x, y) => x + y;
```

x + y



x + y * 2

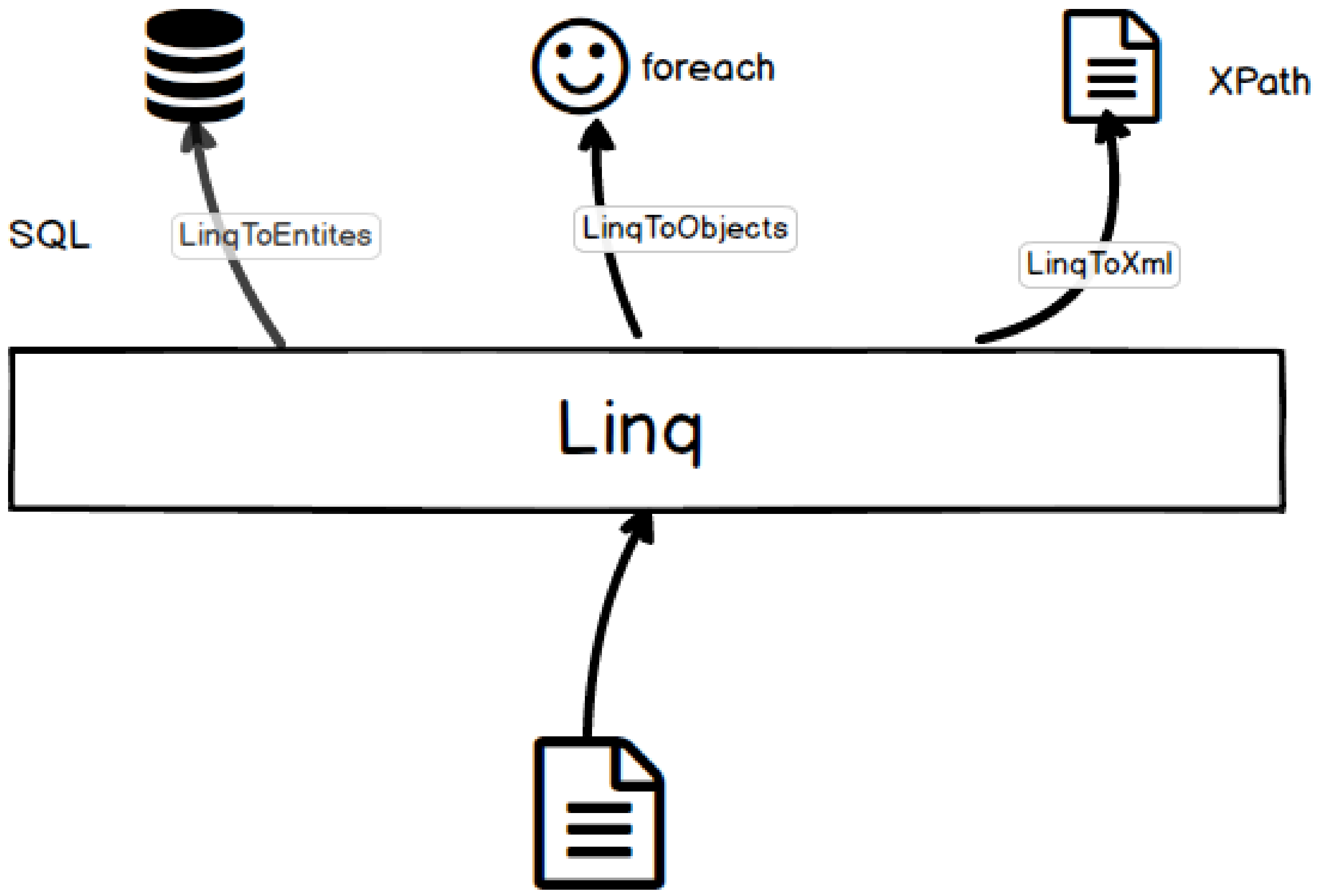




SQL

select Name, Color, UnitPrice from dbo.Products where UnitPrice > 100

deklaratywny

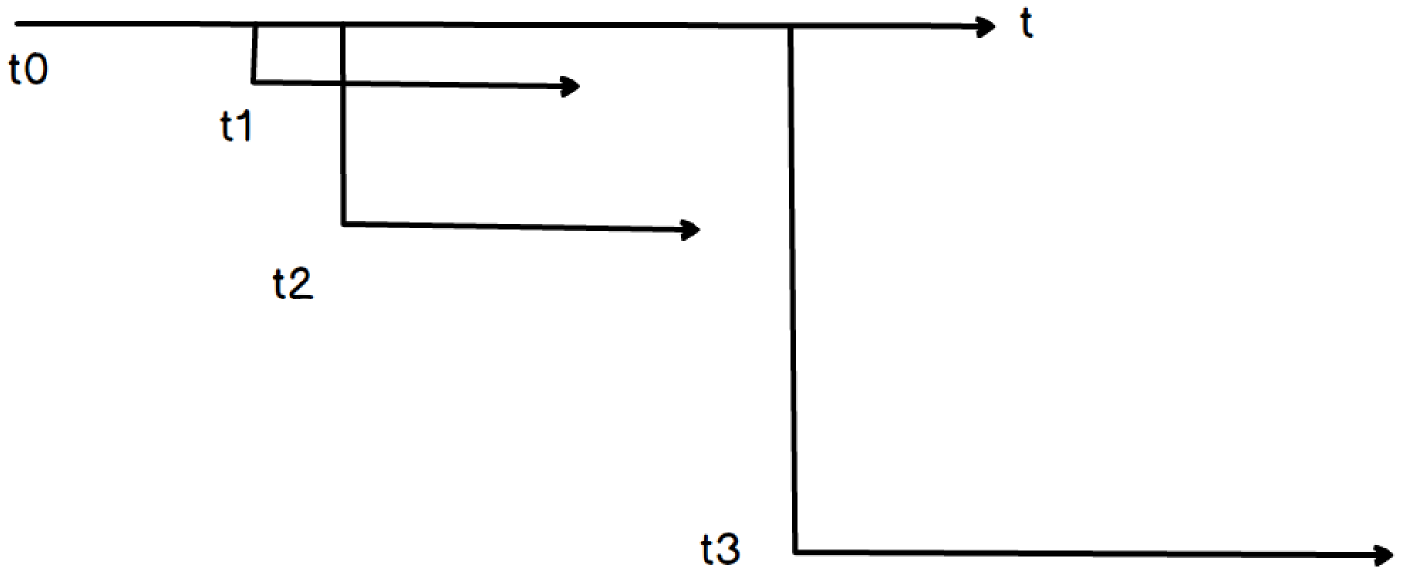

SQL
LinqToEntites
foreach
LinqToObjects
XPath
LinqToXml
Linq

```
ICustomerService customerService = new FakeCustomerService();

ICustomerService customerService = new DbCustomerService(new MyContext());


ICustomerService customerService = Factory.Create(parameter);
```

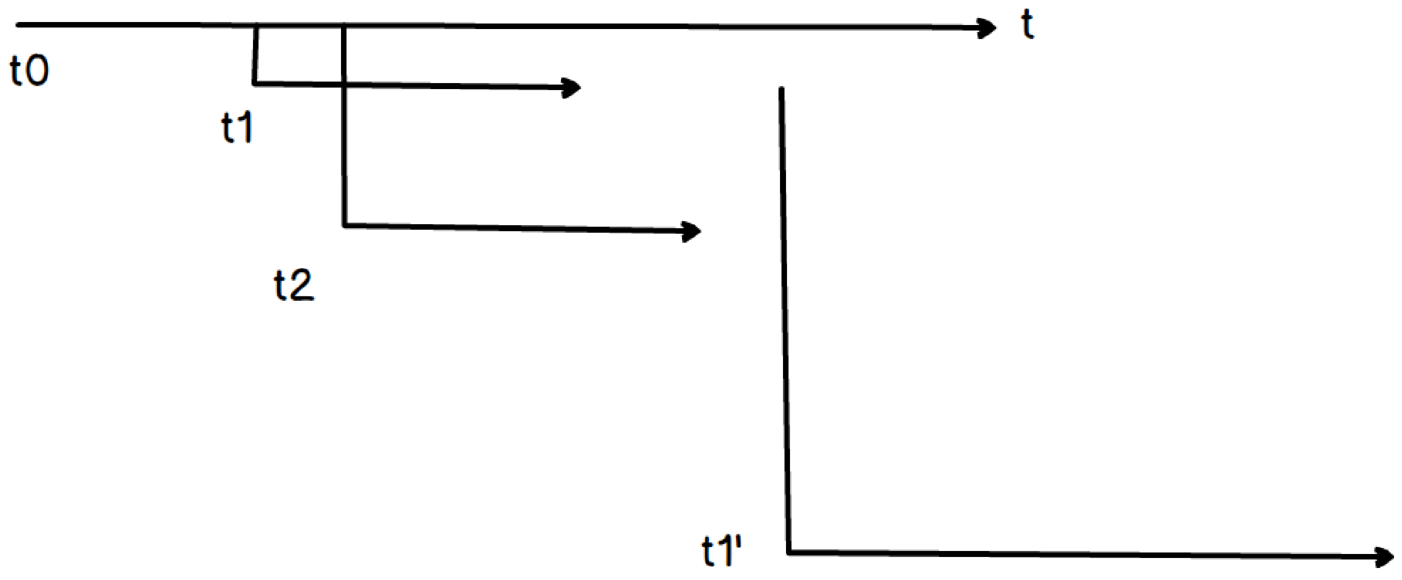
Key	Value	cykl życia
ICustomerService	DbCustomerService	Scoped
MyContext	MyContext	Scoped
IProductService	FakeProductService	Singleton

Kontener (rejestr)

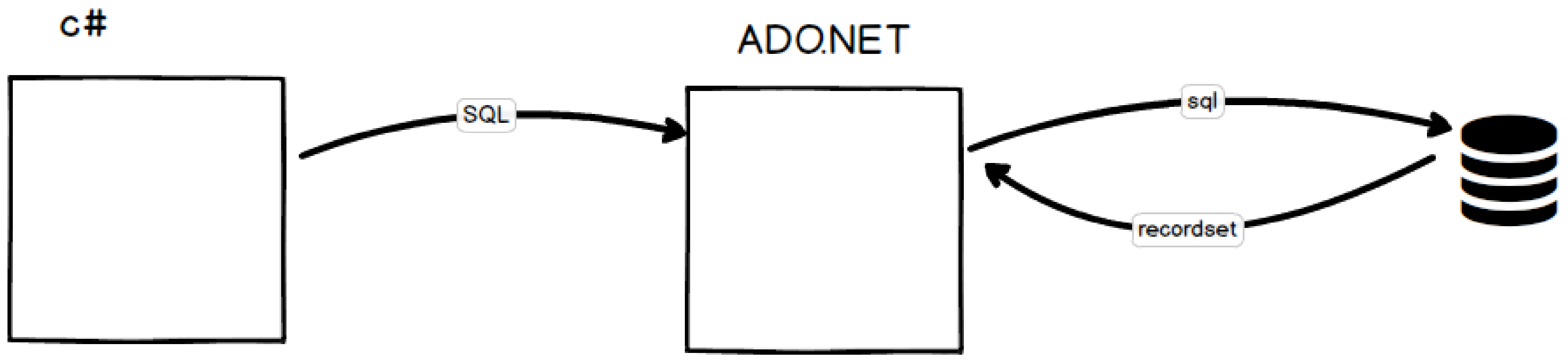
Thread



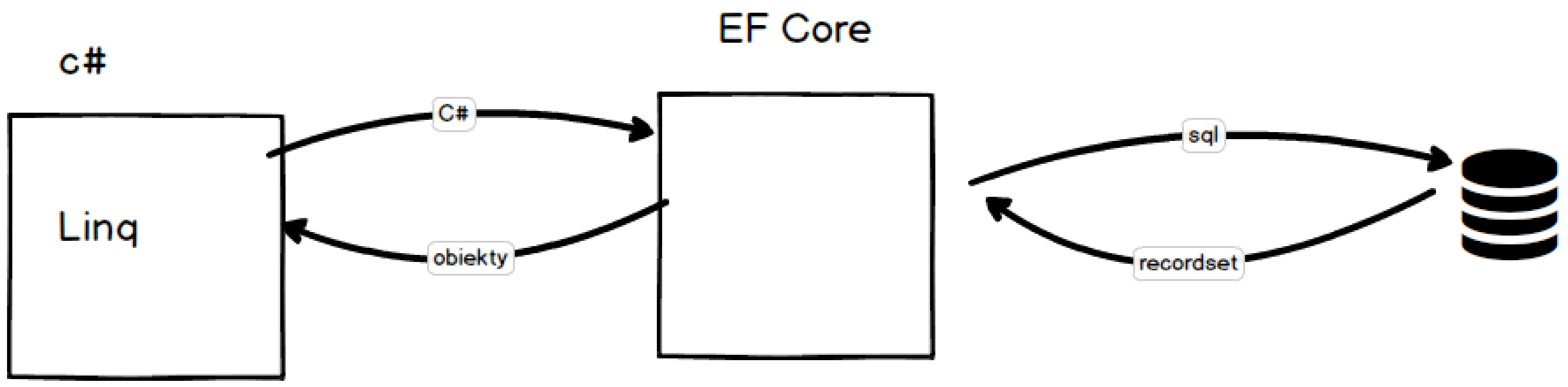
ThreadPool



Task

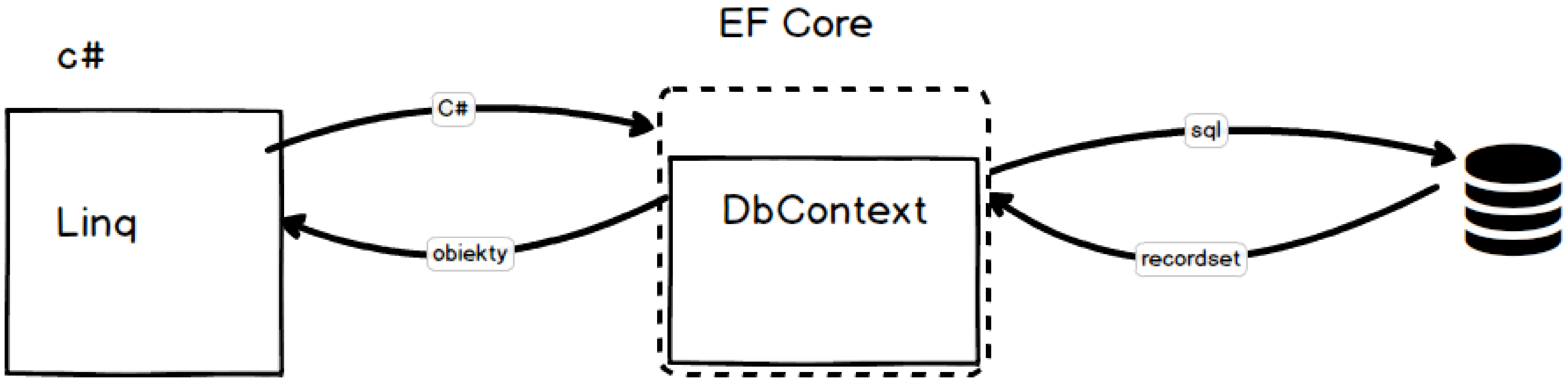


ORM (Object Relation Mapping)



nHibernate

Dapper


EF Core
c#
Linq
C#
obiekty
DbContext
sql
recordset