

---

# Kort om COM, C#, .NET

Disclaimer:

Jag är varken C#, Java eller .NET expert...

# Component Object Model

---

- Microsoft's standard för att möjliggöra interaktion mellan oberoende komponenter
- Ett COM objekt accesseras genom funktioner (metoder) grupperade i ett eller flera gränssnitt
  - Men inte nödvändigtvis alla metoder en klass implementerar
- COM standarden består av
  - En representationsstandard för binära objekt
  - Definition av grundgränssnitt

# COM utveckling

---

- Microsoft office tools for creating documents
  - Print, cut, copy, paste
  - Local software bus to link office tools
  - Dynamic Data Exchange (DDE) - Internal standard, slow, error prone
- Object linking and embedding (OLE)
  - How to create compound documents
  - Using DDE (first)
- Compound Object Model (COM)
  - Local software bus to link general processing components
  - Replaces DDE
- ActiveX of OLE
  - Uses COM (instead of DDE) to define compound documents
  - Provides services
    - Name & property services, event based communication service
    - Document processing specific services (cut, copy, paste, ...)
- Distributed COM (DCOM)
  - Based on Distributed Computing Environment (DCE) + COM
  - Software bus with remote transparency

# COM gränssnitt

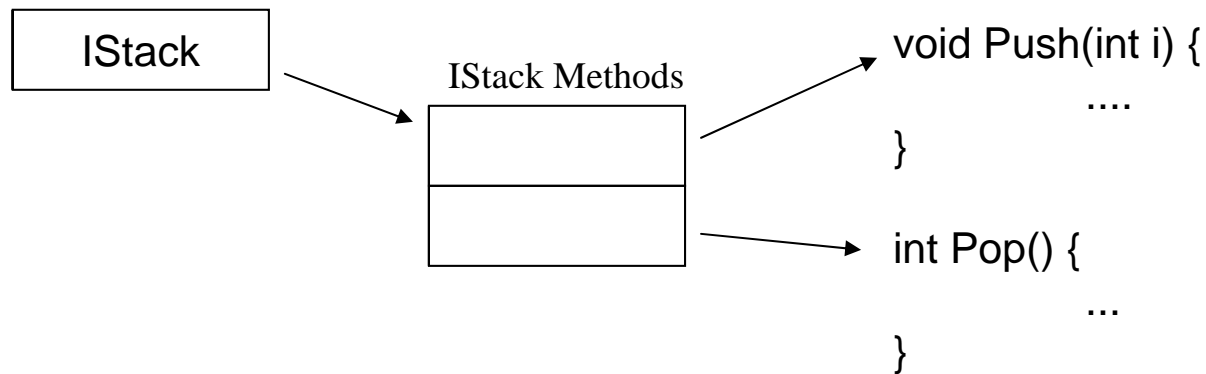
---

- Ett COM-gränssnitt definierar
  - grupper av funktioner vars prototyper är definierade
- Gränssnittsimplementation
  - kod för att förverkliga det gränssnittet specificerar
- Objekt kan innehålla flera olika gränssnitt
  - Dock om ett objekt implementerar ett gränssnitt, måste alla gränssnittets funktioner ha någotslags implementation
- Varje gränssnitt har sitt egen Interface Identifier IID, motsvarar en GUID (Global Unique ID = 128 bit identifierar)
  - Ändrar man gränssnittet, bör även IID ändra

# Instans av en gränssnittsimplementation

---

- I praktiken motsvaras en instans av en gränssnittsimplementation av en pekare
  - Pekaren pekar på en tabell med de funktioner (metoder) gränssnittet implementerar



# Gränssnittet IUnknown

---

- Alla gränssnitt för COM-objekt måste ärvas och implementera IUnknown

```
interface IUnknown {  
    virtual HRESULT QueryInterface(REFIID iid, void  
        **ppvObject) = 0;  
    virtual ULONG AddRef(void) = 0;  
    virtual ULONG Release(void) = 0;  
}
```

# Exempel på användning

---

```
LPLOOKUP *pLookup;
TCHAR szNumber[64];
HRESULT hRes;
    // Call QueryInterface on the component object
    // PhoneBook, asking for a pointer to the ILookup
    // interface identified by a unique interface ID.
hRes = pPhoneBook->QueryInterface(IID_ILOOKUP, &pLookup);

if( SUCCEEDED( hRes ) ) {
    // use Ilookup interface pointer
    pLookup->LookupByName("Daffy Duck", &szNumber);
    // finished using the IPhoneBook interface pointer
    pLookup->Release();
} else {
    // Failed to acquire Ilookup interface pointer.
}
```

# COM klienter och servrar

---

- Klient – en applikation som använder sig av ett COM-objekt genom att kalla på de funktioner en Server erbjuder
- Server – implementerar ett gränssnitt och dess funktionalitet
- Applikationer kan samtidigt fungera som både COM klienter och servrar
  - T.ex. Tabellkalkyl kan läggas in i ordbehandling och vice versa



# COM-gränssnitt och IDL

---

- Gränssnittet beskrivs gärna med MIDL (Microsoft Interface Definition Language),
  - MIDL-kompilatorn ser till att stub-kod genereras som möjliggör att metoder kan kallas så att parametrar överförs på rätt sätt

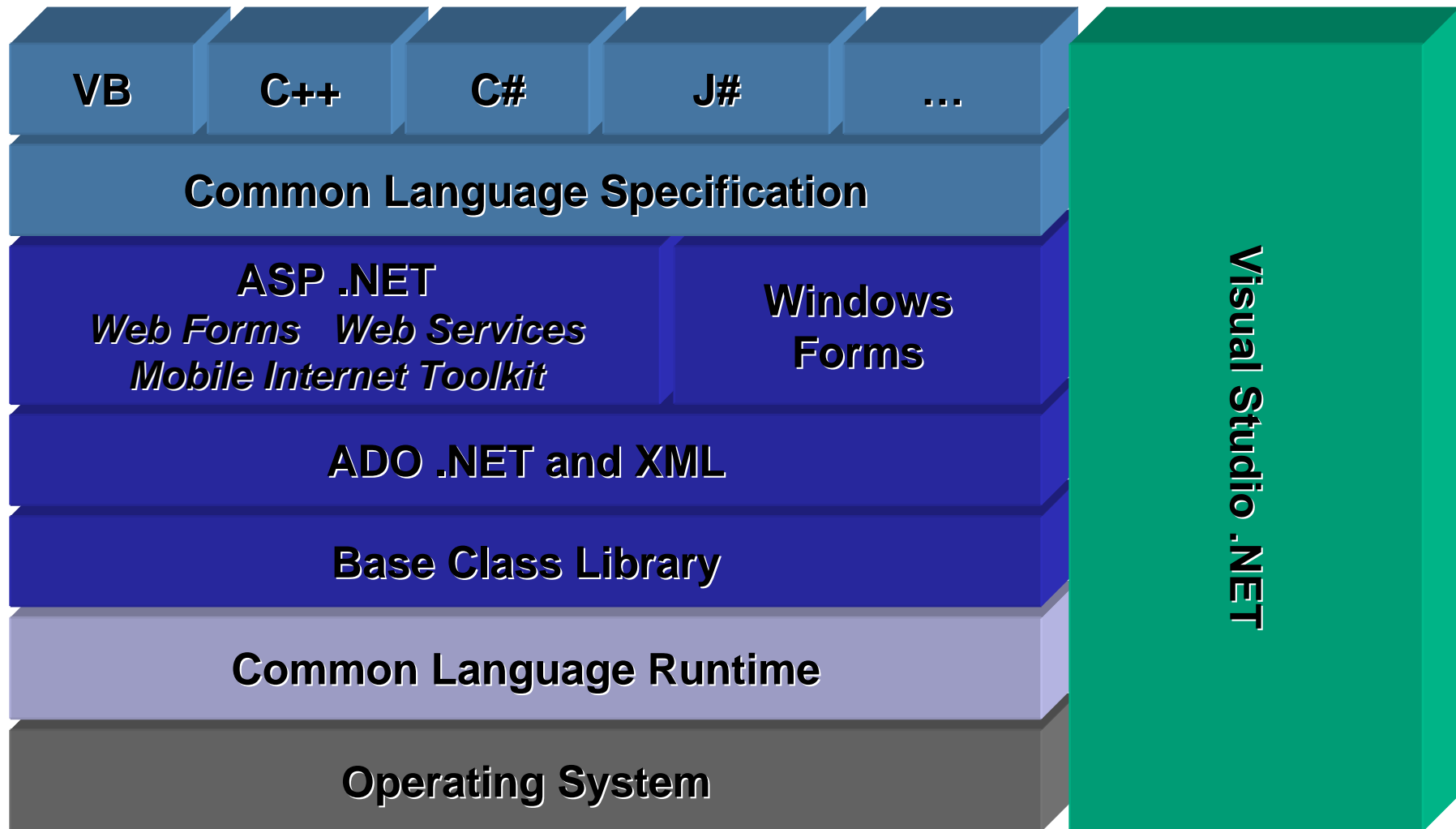
# Vad är .NET?

---

- Svar 1: Microsofts svar på Java
- Svar 2:
  - Common Language Runtime (CLR)
    - Oberoende av programmeringsspråk
      - C#, VB, C++, J#, Python, Cobol, ...
    - Kompileras till ett mellanspråk (intermediate language IL)
  - Just-in-time (JIT) kompilering
    - IL kompileras vid runtime till den maskin koden körs på
  - Minneshantering
    - Garbage collecting, objekt frigörs automatiskt när de inte behövs mera

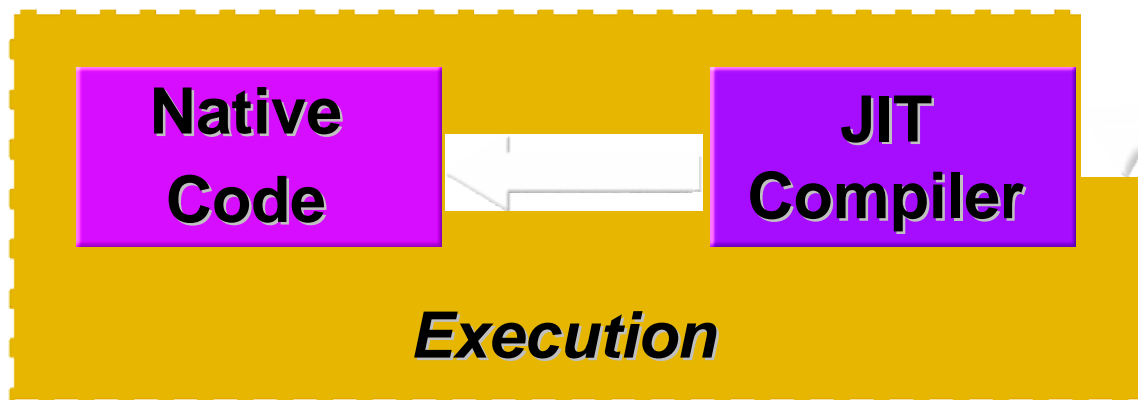
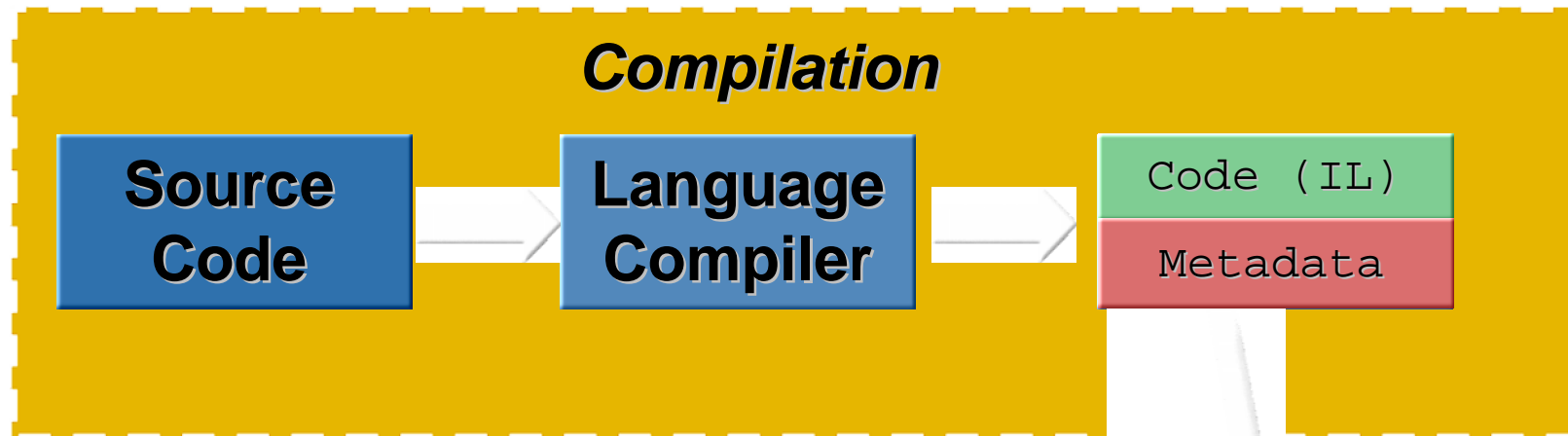
# Implementation and Benefits

## .NET Framework and Tools



# Implementation and Benefits

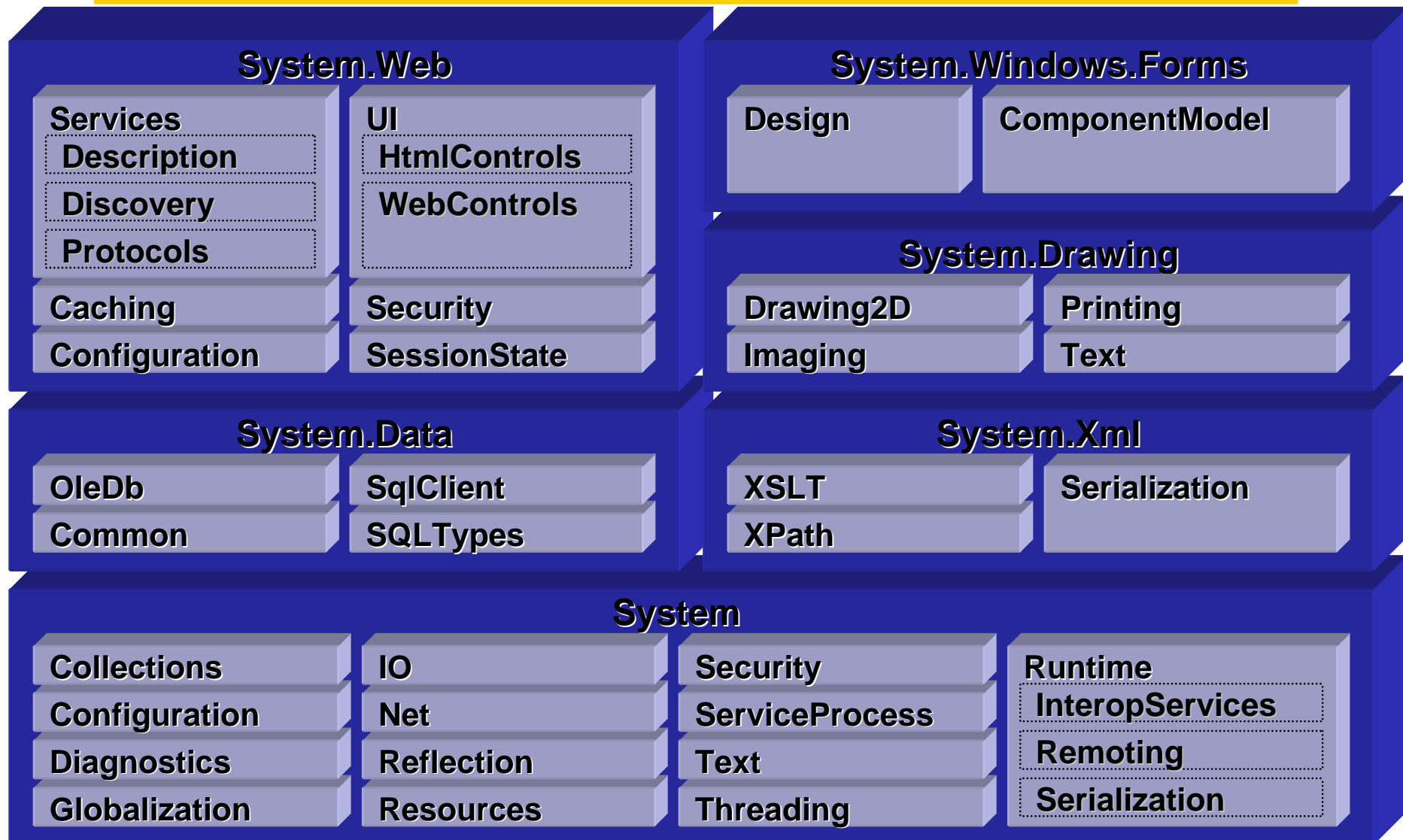
## Compilation and Execution



*At installation or the first time each method is called*

# Implementation and Benefits

## .NET Framework Class Library



# C#

---

- Slides från
  - <http://www.ecma-international.org/activities/Languages/Introduction%20to%20Csharp.ppt>

This document was created with Win2PDF available at <http://www.win2pdf.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.  
This page will not be added after purchasing Win2PDF.