[2012]

|  |
| --- |
| [Team Assignment 03] | Team 02 – The HIT  HIT-Big |



|  |  |
| --- | --- |
| Class K15T2 | Investigation into .NET framework |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Introduction: 3](#_Toc324449089)

[a. For what purpose (domain and/or business context) was the standard created? 3](#_Toc324449090)

[b. Describe the stakeholders: Who created it? Who (domain or community) uses it? How do they use it? 4](#_Toc324449091)

[c. Are there other related and/or competing standards? If so, list them and briefly explain how they differ from the standard you are analyzing. 6](#_Toc324449092)

[2. Description of the Standard: 10](#_Toc324449093)

[a. What styles/patterns does it utilize? Use the terminology presented in the course to describe the architecture. 10](#_Toc324449094)

[b. Are there semantics for using the framework, product or standard? How are they enforced and what are the consequences of violating them? 11](#_Toc324449095)

[c. What variability mechanisms are built into the framework, product or standard – how does it meat different needs of the various stakeholders and/or the business contexts the framework, product or standard might be used in? 12](#_Toc324449096)

[3. Analysis of Properties, Tradeoffs, and Applicability: 12](#_Toc324449097)

[a. Describe the architectural drivers (quality attributes in particular) that the framework, standard, or product is trying to maximize and how these are related to the business context, domain, and/or stakeholder community. 12](#_Toc324449098)

[b. Discuss the structures of the system and explain how the structures support the key goals and architectural drivers as discussed in the previous question. 12](#_Toc324449099)

[c. Can you analyze how well these architectural drivers supported (or not supported) by the artifacts and design descriptions you were able to find? 14](#_Toc324449100)

# Introduction:

## For what purpose (domain and/or business context) was the standard created?

.NET is results of many influences:



The .NET Framework is an integral Windows component that supports building and running the next generation of applications and XML Web services. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

## Describe the stakeholders: Who created it? Who (domain or community) uses it? How do they use it?

**Owned by Microsoft Corporation.**

Microsoft involves the community by actively getting feedback through multiple channels

**Who use it:** Users and Developers.

**How to use it:**

* For Users:

If you don’t develop .NET Framework applications, but you use them, you don’t need to be expert at the .NET Framework or its operation. For the most part, the .NET Framework is completely transparent to users.

If you are using the Windows operating system (such as: Window XP, Window Vista, Window 7), the .NET Framework may already be installed on your computer. In addition, if you install an application that requires the .NET Framework, the application's setup program might install any specific version of the .NET Framework on your computer. In some cases, you may see a dialog box that asks you to install the .NET Framework. If you have just tried to run an application when this dialog box appears and if your computer has Internet access, you can go to a web page that lets you install the missing version of the .NET Framework.

In general, you should not uninstall any versions of the .NET Framework that are installed on your computer, because an application you use may depend on a specific version and may crash if that version is removed. Note that multiple versions of the .NET Framework can be loaded on a single computer at the same time. This means that you do not have to uninstall previous versions in order to install a later version.

* For Developers:
* The .NET Framework provides the following services for application developers:
* Memory management. In many programming languages, programmers are responsible for allocating and releasing memory and for handling object lifetimes. In .NET Framework applications, the CLR provides these services on behalf of the application.
* A common type system. In traditional programming languages, basic types are defined by the compiler, which complicates cross-language interoperability. In the .NET Framework, basic types are defined by the .NET Framework type system and are common to all languages that target the .NET Framework.
* An extensive class library. Instead of having to write vast amounts of code to handle common low-level programming operations, programmers can use a readily accessible library of types and their members from the .NET Framework Class Library.
* Development frameworks and technologies. The .NET Framework includes libraries for specific areas of application development, such as ASP.NET for web applications, ADO.NET for data access, and Windows Communication Foundation for service-oriented applications.
* Language interoperability. Language compilers that target the .NET Framework emit an intermediate code named Common Intermediate Language (CIL), which, in turn, is compiled at run time by the common language runtime. With this feature, routines written in one language are accessible to other languages, and programmers can focus on creating applications in their preferred language or languages.
* Version compatibility. With rare exceptions, applications that are developed by using a particular version of the .NET Framework can run without modification on a later version.
* Side-by-side execution. The .NET Framework helps resolve version conflicts by allowing multiple versions of the common language runtime to exist on the same computer. This means that multiple versions of applications can also coexist, and that an application can run on the version of the .NET Framework with which it was built.
* Multi-targeting. By targeting the .NET Framework Portable Class Library, developers can create assemblies that work on multiple .NET Framework platforms, such as the .NET Framework, Silverlight, Windows Phone 7, or Xbox 360.

You can choose any programming language that supports the .NET Framework to create your application. Because the .NET Framework provides language independence and interoperability, you can interact with other .NET Framework applications and components regardless of the language with which they were developed.

To develop .NET Framework applications or components, do the following:

1. Install the version of the .NET Framework that your application will target. The most recent production version is the .NET Framework 4.
2. Select the type of the Applications or services you want to

***For example,*** you can use the .NET Framework to develop the following types of applications and services:

* Console applications.
* Windows GUI applications (Windows Forms).
* Windows Presentation Foundation (WPF) applications.
* ASP.NET applications.
* Web services.
* Windows services. See Introduction to Windows Service Applications.
* Service-oriented applications using Windows Communication Foundation (WCF).
* Workflow-enabled applications using Windows Workflow Foundation (WF).

1. Select the .NET Framework language or languages that you will use to develop your applications. A large number of languages are available, including Visual Basic, C#, F#, and C++ from Microsoft…
2. Select and install the development environment that you will use to create your applications and that supports your selected programming language or languages. The integrated development environment for .NET Framework applications available from Microsoft is Visual Studio. It is available in a number of retail and free editions.

## Are there other related and/or competing standards? If so, list them and briefly explain how they differ from the standard you are analyzing.

The following sections describe the main components and features of the .NET Framework in greater detail.

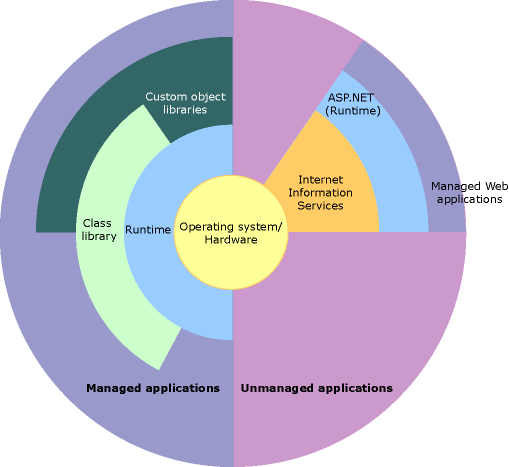
* + 1. **.NET**
* *The .NET Framework* (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages). The .NET library is available to all the programming languages that .NET supports. The .NET Framework is intended to be used by the most new applications development environment largely for .NET software called *Visual Studio*
* *Owned* by Microsoft Corporation, established on April 4, 1975. Microsoft involves the community by actively getting feedback through multiple channels
* *Architecture:* The .NET Framework has two main components: the common language runtime and the .NET Framework class library
* The **Common Language Runtime (CLR),** an application virtual machine that provides important services such as security, memory management, and exception handling.
* The .NET Framework’s **Base Class Library (BLC)**, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) application to application based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services. Programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers.

Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable ASP.NET applications and XML Web services

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents.

Figure 1: Show the relationship between the CLR and the BCL of applications and to the overall system.



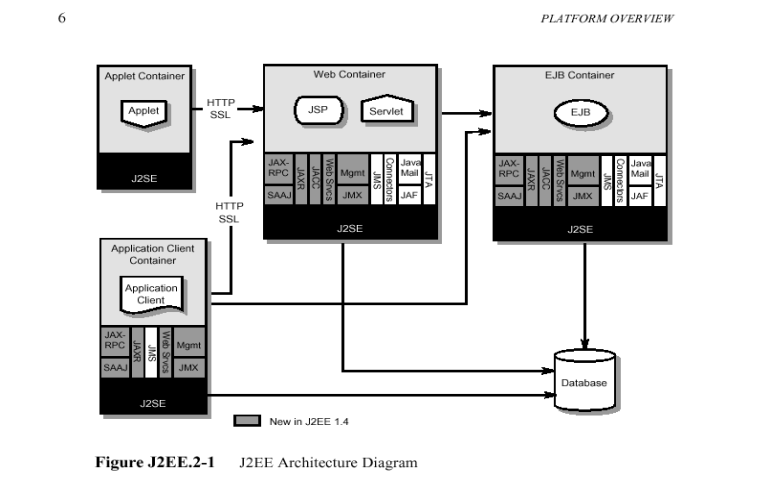
* *Version:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Version Studio** | **Default in Windows** |
| **1.0** | February 13, 2002 | Visual Studio.NET | Windows XP Tablet and Media Center Editions |
| **1.1** | April 24, 2003 | Visual Studio 2003 | Windows Server 2003 |
| **3.0** | November 6, 2006 |  | Windows Vista, Windows Server 2008 |
| **4.0** | April 12, 2010 | Visual Studio 2010 |  |

* + 1. **J2EE**
* *J2EE* short for Java 2 Platform Enterprise Edition is an industry standard specification for building distributed object-oriented enterprise systems written by Java
* *Owned by* Sun Microsystems, Inc. founded on February 24, 1982.

Sun Microsystems, Inc. involves the community in building the specification through the Java Community Process

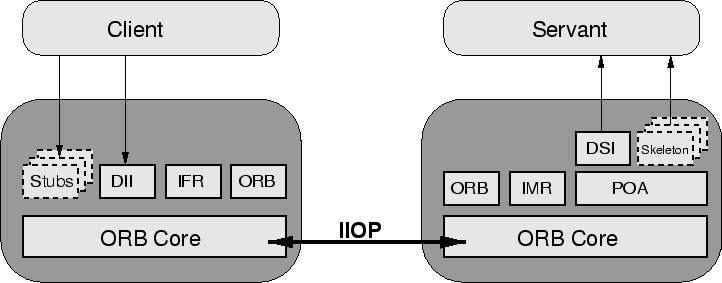
* *Architecture:* shown in Figure 3.



* *Version:*

|  |  |
| --- | --- |
| **Version** | **Release Date** |
| **J2EE 1.2** | December 12, 1999 |
| **J2EE 1.3** | September 24, 2001 |
| **J2EE 1.4** | November 11, 2003 |
| **Java EE 5** | May 11, 2006 |
| **Java EE 6** | December 10, 2009 |

* + 1. **CORBA**
* *COBRA* is acronym for Common Object Request Broker Architecture, OMG’s open vendor-independent architecture and infrastructure that computer applications use to work together over networks. Using the standard protocol IIOP, a CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network.
* *Owned* by OMG. OMG short for Object Management Group has been an international, open membership, not-for-profit computer industry consortium with a membership of more than 700 companies (established since 1989). The organization ‘s goal is to provide a common framework for developing applications using object- oriented programming techniques
* Architecture shown in Figure 5.

**

* CORBA applications are composed of objects, individual units of running software that combine functionality and data. Typically, there are many instances of an object of a single type - for example, an e-commerce website would have many shopping cart object instances, all identical in functionality but differing in that each is assigned to a different customer, and contains data representing the merchandise that its particular customer has selected. For other types, there may be only one instance. When a legacy application, such as an accounting system, is wrapped in code with CORBA interfaces and opened up to clients on the network, there is usually only one instance.
* *Version*

|  |  |  |
| --- | --- | --- |
| **Version** | **Release Date** | **URL** |
| **1.0** | August 1991 | [http://www.omg.org/spec /CORBA/1.0/](http://www.omg.org/spec%20/CORBA/1.0/) |
| **2.0** | February 1997 | [http://www.omg.org/spec /CORBA/2.0/](http://www.omg.org/spec%20/CORBA/2.0/) |
| **2.6** | December 2001 | [http://www.omg.org/spec /CORBA/2.6/](http://www.omg.org/spec%20/CORBA/2.6/) |
| **3.0** | June 2002 | [http://www.omg.org/spec /CORBA/3.0/](http://www.omg.org/spec%20/CORBA/3.0/) |
| **3.1** | January 2008 | [http://www.omg.org/spec /CORBA/3.1/](http://www.omg.org/spec%20/CORBA/3.1/) |

**Compare.NET and Java Architecture**

|  |  |
| --- | --- |
| **.NET Architecture** | **Java Architecture** |
| Designed to support multiple different programming languages. Currently, 30 languages support the .NET architecture. | Though other languages’ code can be converted run under JVM, they don’t acquire true cross-languages capabilities. |
| Compiles the source code to Intermediate Language (II), which is itself a language. | Compiles the source code Java byte code, which by itself is not a language. |
| CLR implements a contiguous memory allocation algorithm. | JVM implements a noncontiguous memory allocation algorithm. |
| Compiles the source code twice during the process of converting to native code. Compiling works faster than interpreting. | Compiles and interprets the source code once during the process of converting it to native code. |

# Description of the Standard:

## What styles/patterns does it utilize? Use the terminology presented in the course to describe the architecture.

The.NET Framework sits on top of the operating system and is made up of two parts, the

* Common Language Runtime (CLR) and the Framework Base Class Libraries (FCL). Each one of these parts plays an important role in the development of .NET applications and services.
* The Common Language Runtime (CLR), an application virtual machine that provides important services such as security, memory management, and exception handling.
* The .NET Framework’s Base Class Library (FCL).object-oriented collection of reusable types.

Steps for executing:

1. Compiling source code => managed modules(IL –Intermediate Language IL – CPU independent + metadata)
2. Combining managed modules and resource files into assemblies (EXE or DLL) Metadata also included in assemblies
3. Compiling (JIT) assemblies + FCLs into native machine codes.

So the patterns of .NET framework:

* Component – based design .EXE + 1 or more .DLLs.
* N-tiers, N-layers design: .NET framework is on the top of OS, including CLR (runtime environment to execute code) and FCL (collection of reused classes of many different kinds).Source code is written based on FCL (Figure 1.)
* SQA (Service oriented architecture)

## Are there semantics for using the framework, product or standard? How are they enforced and what are the consequences of violating them?

|  |  |  |
| --- | --- | --- |
| **Semantics** | **How are they enforced** | **Consequences of violating** |
| **Clients need CLR & FCL to run** | * Available via Redistributable .NET Framework * Runs on 98 and above, NT windows and above | Can’t run |
| **.NET languages: C#, Basic, Jscript, Fortran, Java#...,and any language that will target CLR** | Language Compilers that compile the source codes into MIL (Microsoft Intermediate Language, independent with CPU) | Can’t run |

## What variability mechanisms are built into the framework, product or standard – how does it meat different needs of the various stakeholders and/or the business contexts the framework, product or standard might be used in?

|  |  |
| --- | --- |
| **Variability mechanisms** | **Needs of the various stakeholders** |
| Support many kinds of applications and services: Console, Windows GUI, ASP.NET, Web services, Windows services, Service-oriented, Workflow- enabled application… | Can meet the different needs of stakeholders: Console, Web page, Windows form, connected systems… |
| Support languages C#, Basic, Jscript, Fortran, Java#...The third party can create any language that will target CLR | They can choose any language among .NET languages they like |

# Analysis of Properties, Tradeoffs, and Applicability:

## Describe the architectural drivers (quality attributes in particular) that the framework, standard, or product is trying to maximize and how these are related to the business context, domain, and/or stakeholder community.

|  |  |  |
| --- | --- | --- |
| **No.** | **Quality Attributes** | **Concern related to...** |
| **1** | Scalability | Variations in load without human intervention |
| **2** | Availability | 24/7 availability with very small downtime periods |
| **3** | Security | Authenticate users and protect against data |
| **4** | Usability | Different users should be able to access different contents in different forms |
| **5** | Performance | Responsive system |
| **6** | Portability | Minimal work on a variety of computing platform |
| **7** |  | Developers should be provided with facilities to manage common services as transactions, name services ,and security |
| **8** | Interoperability | Support interoperation of server –side components implemented on different vendor implementations; allow bridges for interoperability of the J2EE platform to other technologies such as CORBA and Microsoft component technology |
| **9** | Extensibility | Allow incorporation of relevant new technologies as they are developed. |

## Discuss the structures of the system and explain how the structures support the key goals and architectural drivers as discussed in the previous question.

|  |  |  |
| --- | --- | --- |
| **No.** | **Quality Attributes** | **Rationale** |
| **1** | Scalability | The runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting. |
| **2** | Availability | J2EE-compliant systems provide ready-to-use transaction services that enhance availability and reliability of the application by providing built-in failure recovery mechanisms  The runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting. |
| **3** | Security | Built-in Security services and can be combined with security services of Windows, IIS, SQL Server |
| **4** | Usability | .NET technologies as ASP.NET, WCF, WF… that enable the reading of content to suit different users |
| **5** | Performance | Compiles the source code twice during the process of converting to native code. Compiling works faster than interpreting.  The runtime is designed to enhance performance. Although the CLR provides many standard runtime services, managed code is never interpreted. Just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager further improves performance.  Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting. |
| **6** | Portability | Cross-platform execution realized in two ways:   * Apps are written against Framework Class Library (FCL), not underlying OS * Compilers generate generic assembly language which must be executed by the Common Language Runtime (CLR) compilers gen. |
| **7** | Buildability | The .NET Framework includes libraries for specific areas of application development, such as ASP.NET for web applications, ADO.NET for data access, and Windows Communication Foundation for service-oriented applications…It provides large library to reuse.  IDE, Visual Studio, provides many tools to develop. |
| **8** | Interoperability | Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.  Language interoperability. Language compilers that target the .NET Framework emit an intermediate code named Common Intermediate Language (CIL), which, in turn, is compiled at run time by the common language runtime. With this feature, routines written in one language are accessible to other languages, and programmers can focus on creating applications in their preferred language or languages. |
| **9** | Extensibility | Component-based approach allows for future extension (code reuse, independent deploying, updating)  n-tiers approach allows the new technologies, frameworks to be included in the top |

## Can you analyze how well these architectural drivers supported (or not supported) by the artifacts and design descriptions you were able to find?

|  |  |
| --- | --- |
| **Artifacts** | **How well** |
| <http://en.wikipedia.org/wiki/.NET_Framework> |  |
| <http://en.wikipedia.org/wiki/J2EE> |  |
| <http://en.wikipedia.org/wiki/Sun_Microsystems> |  |
| <http://en.wikipedia.org/wiki/Microsoft> |  |
| <http://edndoc.esri.com/arcobjects/9.1/ArcGISDevHelp/DevelopmentEnvs/DotNet/Introduction.htm> |  |
| <http://www.informit.com/articles/article.aspx?p=24924&seqNum=4> |  |
| “Software Architecture in Practice” by Len Bass, 2003 |  |
| <http://www.omg.org/gettingstarted/corbafaq.htm> |  |
| <http://msdn.microsoft.com/en-us/library/aa730858(v=vs.80).aspx> |  |
| <http://msdn.microsoft.com/en-us/library/ee658094.aspx#Availability> |  |
| <http://msdn.microsoft.com/en-us/library/bb402962> |  |
| <http://apparchguide.codeplex.com/wikipage?title=Chapter%204%20-%20Designing%20Your%20Architecture> |  |
| <http://msdn.microsoft.com/en-us/library/zw4w595w.aspx> |  |
|  |  |
|  |  |