**Part 3 – Analysis of Properties, Tradeoffs, and Applicability:**

Analyze the standard with respect to the following:

* 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.
* Discuss the structures of the system and explain how the structures support the key goals and architectural drivers as discussed in the previous question.
* Can you analyze how well these architectural drivers supported (or not supported) by the artifacts and design descriptions you were able to find?

**Answer:** 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.

|  |  |
| --- | --- |
| QA | Concern related to |
| Scalability | System should support variations in load without human intervention |
| Availability | System should provide 24/7 availability with very small downtime periods |
| Security | System should authenticate users and protect against unauthorized access to data |
| Performance | Users should be provided with responsive systems |
| Interoperability | Support interoperation of server-side components implemented on different vendor implementations; allow bridges for interoperability of the Language .NET Framework platform to other technologies |
| Buildability |  |
| Portability | Create compatible implementations of the framework and its languages on other platforms. |

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

|  |  |
| --- | --- |
| QA | Rationale |
| Scalability | * .NET framework multi-tiered architecture and component-based .NET framework architecture has built-in mechanisms for expanding the number of servers available in a configuration and to load balance among server |
| Availability | * .NET Framework provide ready-to-use transaction services that enhance availability and reliability of the application by providing built-in failure recovery mechanisms |
| Security | * .NET Framework provide declarative, role-based security mechanisms and programmatic security mechanisms that are ready to use   The design is meant to address some of the vulnerabilities, such as buffer overflows, which have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.  The .NET Framework provides several mechanisms for protecting resources and code from unauthorized code and users:   * ASP.NET Web Application Security provides a way to help limit faccess to a site by comparing authenticated credentials (or representations of them) to Microsoft Windows NT file system permissions or to an XML file that lists authorized users, authorized roles, or authorized HTTP verbs. * Code access security uses permissions to help limit the access that code has to protected resources and operations. It helps protect computer systems from malicious mobile code and helps provide a way to allow mobile code to run safely. (Code access security, together with the policies that govern it, are referred to as evidence-based security.) * Role-based security provides information needed to make decisions about what a user is allowed to do. These decisions can be based on either the user's identity or role membership, or both. |
| 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 language * Because computer systems commonly require interaction between newer and older applications, the .NET Framework provides means to access functionality implemented in programs that execute outside the .NET environment. Access to COM components is provided in the System.Runtime.InteropServices and System.EnterpriseServices namespaces of the framework; access to other functionality is provided using the P/Invoke feature. |
| Buildability | * .NET application servers provide many ready-to-use services for building server-side C#,… applications, including transactions, persistence, threading, and resource management; developer is thus freed from low-level distribution details; Microsoft provides a reference .NET implementation; application server vendors also participate in the .NET framework specification process |
| Portability | * While Microsoft has never implemented the full framework on any system except Microsoft Windows, the framework is engineered to be platform agnostic, and cross-platform implementations are available for other operating systems (see Silverlight and the Alternative implementations section below). Microsoft submitted the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language), the C# language, and the C++/CLI language to both ECMA and the ISO, making them available as official standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms. |

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