Server Technology – Web Based Service Oriented Architecture for Mobile Augmented Reality System

Jatin Shah
Research Scholar, KSV University,Gandhinagar, Gujarat,India

DR. Bijendra Agrawal
Director, VJKM Groups of Institution, Vadu, Kalol, Gujarat

Abstract:

Server Technology stands for lots of technology in mind like Microsoft, Sun Java, IBM, Open Source and many more. In mobile augmentation, server plays very important role to augment the data. Responsibility of the server is to collect the data , mixed virtual data with real data and these data sent back to client on Remote device at Remote place

In this paper we briefly discuss about the server technology for web based Service oriented, also the processing software required for augmentation, it's software technology, how they accept input from various types of devices and generated output data of various types like audio, video, 3-D graphics.

Keywords: Server Technology, .NET, Java

Introduction :

In Mobile Augmented Reality System , the mainly process in three steps – first the data generation unit which tracks the data for it's location and therefore it's surrounding environments, second the data is send for the augmentation so finding the processing server as per requirement (data are various different types like audio, video, Image, 3-D etc) and after that augmented data sent back, and Third is the displaying the data on client device. It's possible that the data generation , displaying the data will be the same device or any other device at remote location. So here the main theme is augmentation, which requires so large processing capacity , memory, power and storage capacity[in Ghz].

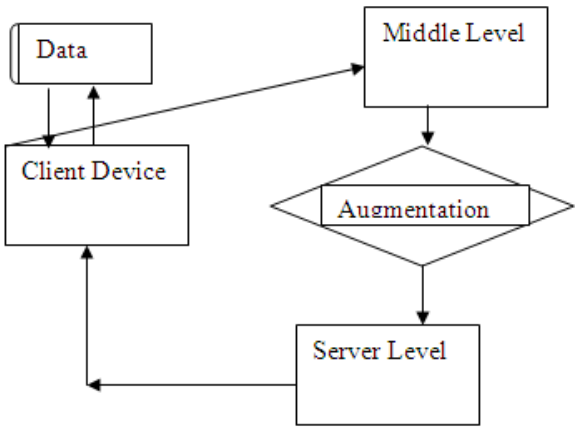
So the augmentation occurs at the place is called server and the technology means we use different software's for managing different types of data for process so it's called Server technology. In server we requires the different software for processing audio , video or mixed data, also the Server OS, platform for which all software and data is resides for processing.

Design :

Design Architecture for server is responsible for accept data from mobile web services . and storing data for processing, so caching mechanism is required at server side. Server continuous accept data from remote services and process this data at the same time parallel and sent back this processing data to remote client. Storing of remote client address is also important so server can easily find the client for data back.

Main responsibility of the server is to process data continuously as per their type like audio, video etc. so at remote place the client can choose the server type for which type of data it can handle so the process will run continue and augmented data can view by the remote client.

Client side web Based Service is responsible to easily access on client browser , so client can register themselves to authorized for access other required services. Main responsibility of the client is send data to the middle level services to reach the destination (linked with other web based services – n level) and receive the augmented data from directly from server side client or middle level services.



Develop Web Based Service Oriented Architecture for client using any of the web tools because Web Services has an open standard so any device for client as well as server side client can access it without any problem of OS or with minimum Installation or Processes. Authorized user have given different rights to send video, audio or 3D image data to middle level services.

Here we define the terms what is Augmented Reality and Mobile Augmented Reality System

"The Computer System that combined the real and the virtual in order to assist users in interacting with their physical environments are called Augmented Reality System. "

"Mobile Augmented Reality System is one in which augmentation occurs through available knowledge of where the user is e.g. user's location and therefore his surrounding environments.'

Web Technologies for Client :

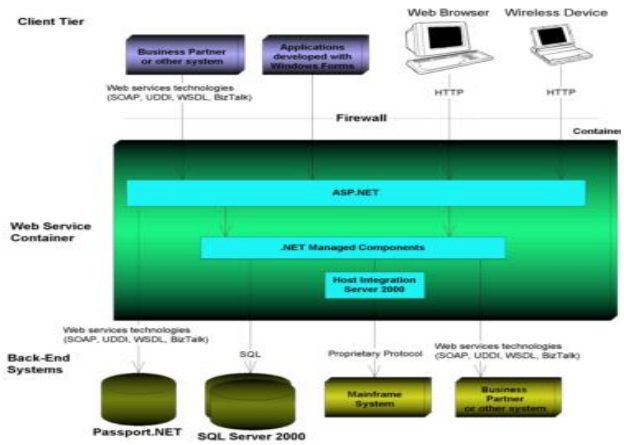
Various types of tools are available in market for developing web based application/software. Like .NET – Microsoft Technology provides

very powerful framework and very rich library to user for any kind of web/window based development for mobile or normal web based application. Java is another platform for developing this kind of application. Java provides us very large library and different packages for this. PHP and ROR are very popular open source web development technology recently use in the market. They also provide the framework and rich library for the advance development.

Due to Mobile OS, it's processing capacity and limited memory and storage space , we focus mainly two technologies .NET and Java. So development should be very easier , provide very powerful languages like ASP.NET , J2ME and provide dynamic controls that can help to user faster development and interacting with Real Time Audio, Video, or Image Processing that requires for Augmentation.,

Microsoft .NET :

Microsoft .NET is product suite that enables organizations to build smart, enterprise-class web services. Note the important difference :.NET is a product strategy, whereas J2EE is a standard to which products are written. Microsoft .NET is largely a rewrite of windows DNA, which was Microsoft's previous platform for developing enterprise applications. Windows DNA includes many proven technologies that are in production today, including Microsoft Transaction Server (MTS) and COM+, Microsoft Message Queue technologies, and includes a web services layer as well as improved language support.



The .NET application is hosted within a container, which provides qualities of service necessary for enterprise applications, such as transactions, security , and messaging services.

The Business layer of the .NET application is built using .NET managed components. This layer performs business processing and data logic. It connects to databases using Active Data Objects (ADO.NET) and existing systems using services provided by Microsoft Host Integration Server 2000, such as the COM Transaction Integrator (COM TI). It can also connect to business partners using Web Services Technologies (SOAP, UDDI, WSDL).

Business Partners can connect with the .NET application through web services technologies.

Traditional 'thick' clients, web browsers, wireless devices connect to Active Server Pages (ASP.NET) which render user interfaces in HTML, XHTML, or WML.

.NET Framework : Microsoft .NET offers language-independence and language-interoperability. This is one of the most intriguing and fundamental aspects of the .NET platform. A single .NET component can be written, for example, partially in VB.NET, the .NET version of Visual Basic, and C#, Microsoft's new object oriented programming language. How does this work ? first, source code is translated into Microsoft Intermediate Language, sometimes abbreviated MSIL, sometimes IL. This IL code is language-neutral, and is analogous to Java bytecode. The IL code then needs to be interpreted and translated into a native executable. The .NET Framework includes the Common Language Runtime (CLR), analogous to the Java Runtime Environment (JRE), which achieves this goal.

The CLR is Microsoft's intermediary between .NET developer's source code and the underlying hardware, and all .NET code ultimately runs within the CLR. This CLR provides many exciting features not available in earlier versions of Windows DNA, such as automatic garbage collection, exception handling, cross-language inheritance, debugging, and "side-by-side" execution of different versions of the same .NET component.

JAVA :The Foundation for J2EE

The J2EE architecture is based on the Java programming language. What's exciting about Java is that it enables organizations to write their code once , and deploy that code onto any platform. The process is as follows :

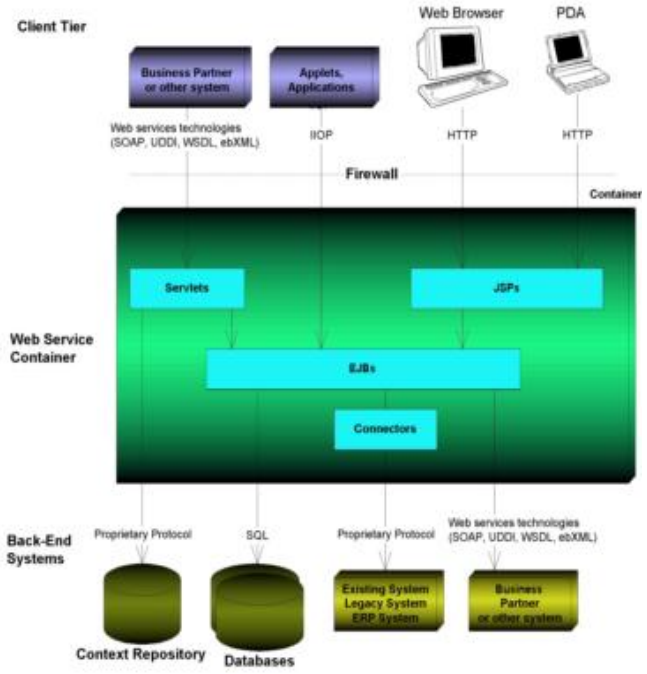
1. Developers write source code in Java.
2. the Java code is compiled into bytecode, which is a cross-platform intermediary, halfway between source code and machine language.

3. when the code is ready to run, the Java runtime Environment (JRE) interprets this bytecode and executes it at run-time.

4. J2EE in an application of Java. J2EE components are transformed into bytecode and executed by a JRE at runtime. Even the containers are typically written in Java.

J2EE and Web Services

J2EE has historically been an architecture for building server-side deployments in the Java programming language. It can used to build traditional web sites, software components, or packaged applications. J2EE has recently been extended to include support for building XML-based web services as well. These web services can interoperate with other web services that may or may not have been written to the J2EE standard.



J2EE application is hosted within a container, which provides qualities of service necessary for enterprise applications, such as transactions, security, and persistence services.

The business Layer performs business processing and data logic. In large-scale J2EE applications, business logic is built using Enterprise JavaBeans (EJB) components. This layer performs business processing and data logic. It connects to databases using Java Database Connectivity (JDBC) or SQL/J, or existing systems using the Java Connector Architecture (JCA). It can also connect to business partners using web services technologies (SOAP, UDDI, WSDL, ebXML) through the Java APIs for XML (the JAX APIs).

The business partners can connect with j2EE applications through web services technologies (SOAP, UDDI, WSDL, ebXML). A servlet, which is a request/response oriented Java Object, can accept web service requests from business partners. The servlet uses the JAX APIs to perform web services operations. Shared context services will be standardized in the future through shared context standards that will be included with J2EE.

Traditional 'thick' clients such as applets or applications connect directly to the EJB layer through the Internet Inter-ORB Protocol (IIOP) rather than web services, since generally the thick clients are written by the same organization that authored J2EE application, and therefore there is no need for XML-based web service collaboration.

Web browsers and Wireless devices connect to JavaServer Pages(JSPs) which render user interfaces in HTML, XHTML, or WML.

Analogy : Java and .NET

Feature	J2EE	.NET
Type of technology	Standard	Product
Middleware Vendors	30+	Microsoft
Interpreter	JRE	CLR
Dynamic Web Pages	JSP	ASP.NET
Middle-Tier Components	EJB	.NET Managed Components
Database Access	JDBC,SQL/J	ADO.NET
SOAP,WSDL,UDDI	Yes	Yes
Implicit middleware (load-balancing, etc)	Yes	Yes

Comparative Analysis :

Both Sun J2EE and Microsoft .NET provide runtime mechanisms that insulate software developers from particular dependencies. In addition to a web service-oriented XML layer of indirection between platforms, languages, and enterprise architectures, Sun J2EE and .NET offer language-level intermediation via the Java Runtime Environment (JRE) and the Common Language Runtime (CLR) respectively.

Microsoft .NET offers a variety of time-to-market features not found in J2EE as well. Most notably, ASP.NET is independent of client device, and allows for user interfaces to be rendered to alternative user interfaces without rewriting code. Microsoft also offers Queued

Components which are superior to MessageDriven Beans. It should be noted here that Microsoft has tried to simplify server-side programming greatly by

removing support for features found in traditional enterprise applications, such as stateful servers and simple transactions. If developers need to go outside this box, their code must be made to be non-managed and reside outside the .NET framework rather than take advantage of it. Microsoft also provides business process management and E-Commerce capabilities, which are available in some J2EE implementations but not all.

In conclusion, we feel the ability to achieve rapid application development offered by both J2EE and .NET is definitely not equal. It is, however, comparable. The feature differences are minor and it is very difficult to make a compelling argument either way.

When building web services, in general you should always prefer to have a single-vendor solution. A single vendor is usually more reliable , interoperable and less error-prone than a two-vendor bridged solution.

One of J2EE's strengths is that it has spawned a wide variety of tools, products and applications in the marketplace, which provide more functionality in total than one vendor could ever provide. However , this strength is also a weakness. With lower-end J2EE implementations, you need to mix and match tools to get a complete solution, which could result in low-level hacking due to imperfections in portability. Larger vendors, such as IBM, ORACLE, BEA and iPlanet, each offer a

complete solution, and thus their products are fully interoperable. Thus it is important to choose a larger vendor to avoid interoperability headaches.

.NET provides a fairly complete solution from a single vendor-Microsoft. This solution may lack some of the higher end features that J2EE solutions offer, but in general, the complete web services vision that Microsoft will be providing is equal in scope to that of a larger J2EE vendor.

Developers of Java GUI applications will find that .NET provides a rich set of features in Windows Forms. These features offer an easy migration path for Swing and Java/AWT developers to move graphical applications to the native Windows interface. JSP programmers will find much to like when migrating code to ASP.NET (also known as Web Forms). This improved version of ASP now supports development in C# in addition to existing Visual Basic and Jscript. ASP.NET server controls give developers the flexibility to extend tag funct ionality-perfect for migration of JSP tag libraries. A key differentiator of the .NET platform is integrated support for Web services. As much of .NET is built using XML technologies, the platform offers full support for SOAP, WSDL, and UDDI. With tools such as Microsoft Visual Studio.NET, it is a trivial matter to expose application functionality as a Web service. Although Java does offer all the technologies necessary to build Web services, it is not at the crux of the J2EE platform – this shows in the level of development effort required.

One approach for migrating applications to .NET is to rewrite existing functionality in C#. A total rewrite may be viable if the existing Java application has architectural deficiencies that require a major overhaul to rectify. Development managers should allocate sufficient time for team members to learn the extensive array of .NET API calls, switch development environments, and implement new deployment procedures. "Although the learning process takes time", states O'Brien, "experience has shown development using the .NET environment is more productive than J2EE". A safer strategy would be to migrate parts of the application to .NET and leave the remaining components running in Java. There are multiple ways to accomplish this goal, and the desired approach depends on the level of integration desired. A simple approach is expose key application interfaces using Web services. The .NET components can make SOAP calls to the Web services and incorporate the results. This solution may be viable if, initially, the business logic stayed on the Java server while presentation layer functions are moved over to ASP.NET. Over time the business logic components could be rewritten in C# and deployed on the .NET server. Interoperability between C# and Java code can also be achieved by using wrapper solutions. Wrappers, otherwise known as software bridges, allow for transitions between incompatible software environments. This approach has been used for many years to solve compatibility problems – from *thunking* between Windows and DOS* to connecting mainframes with Web applications. A third party wrapper product, such as JNBridge*, allows .NET classes to call Java classes if as they were native .NET code.

Java applications already deployed on Visual J#, Microsoft's version of Java based on JDK1.1.4, will experience seamless interoperability as both the Java and C# components execute in the same .NET Common Language Runtime. Visual J# is provided by Microsoft primarily as a convenient way for Visual J++ 6.0 developers to adopt the .NET platform. Developers of J2EE applications will not find Visual J# suitable due to the many incompatible changes that have occurred between JDK1.1.4 and the most recent JDK, version 1.3. Microsoft is working to smooth the migration path with the introduction of its Java Language Conversion Assistant* (JCLA) tool. This tool is central to its JUMP to .NET strategy (Java User Migration Path to Microsoft .NET) and is designed to convert Visual J++ 6.0 projects directly into C# with minimal developer intervention. The tool also remaps Java API calls into functionally equivalent .NET versions. Although useful for Visual J++ 6.0 users, JCLA is not suitable for J2EE app locations as it is limited to JDK1.1.4 compliant applications.

ArtinSoft, developers of the Microsoft JCLA tool, is building a conversion tool tailored to the requirements of enterprise Java applications. This tool supports the latest JDK and an ever-increasing swath of the J2EE API. "Users of the Microsoft JCLA tool are experiencing conversion rates of 90%", says Federico Zoufaly, Executive VP of ArtinSoft. "The J2EE focused migration tool – which is in the final development stages – aims to achieve a conversion rate of at least 80%," he added.

Best Technology For Mobile :

Choose write technology for Mobile is the key elements for Mobile Augmented Reality System. We have describe all the factors for mainly this two technology – Microsoft and Java. Developing Web Based Service Oriented Client for Mobile we choose the Microsoft Technology because of so many advantages like portability, scalability, simple Interface etc. for Augmentation we need the client can communicate with the middle level services or direct communicate with the server to send/receive data. For all this we need a powerful server that can process and augment the data and get back to the client. The process goes n level as per requirement. So we need a powerful server that can manage all the task and Microsoft has the best option for giving any kind of server support. So if the client is based on Microsoft that can easily interact with the server or mid level component.

References :

1) http://www.middleware-company.com

2) Easton, M., King, J., Cross Platform .NET Development Using Mono, portable.NET, and Microsoft .NET, Apress,2004

3) Chappel, D., Understanding .NET: A Tutorial and Analysis, Addison-Wesley, 2002

4)Sierra, K., Bates, B., Sun Certified Programmer & Developer for Java 2, McGraw Hill, 2003

5) Meijer, E., Gough, J., Technical Overview of the Common Language Runtime,2001

6) www.technial-insight.com,

Allan McNaughton-Principal Analyst