

روشها و استانداردهای
یکپارچه سازی عرضه خدمات الکترونیکی دستگاه ها
با درگاه خدمات الکترونیک ایران

معاونت فناوری اطلاعات ، وزارت ارتباطات و فناوری اطلاعات

تهیه و تنظیم:

تیم فنی درگاه الکترونیکی ایران

شماره سند: STD-001-01

تاریخ تنظیم: ۲۵ مهر ۱۳۸۶

نگارش: پیش نویس

فهرست

۳	مقدمه
۴	سر آغاز
۴	معرفی معماری سرویس گرا
۴	ویژگیهای معماری سرویس گرا
۵	تکنولوژیهای یکپارچه سازی
۵	تکنولوژی پیام رسانی (Messaging)
۵	تکنولوژی وب سرویس
۶	تکنولوژی CORBA
۷	تکنولوژی گذرگاه مشترک سرویس (ESB)
۸	نتیجه گیری
۹	منابع و ماخذ

مقدمه

هدف از طراحی و ایجاد درگاه خدمات الکترونیک ایران، تسهیل و تسریع در ارائه خدمات به عموم مردم و حرکت به سمت و سوی ایجاد دولت الکترونیک واحد می‌باشد. در این راستا لازم است تمهیداتی اندیشیده شود تا درگاه مذکور قادر به تعامل و برقراری ارتباط با سامانه‌ها، بانکهای اطلاعاتی و زیر ساختهای موجود در سازمانها و ارگانهای دولتی به منظور اخذ خدمات و ارائه متمرکز آنها به کاربران باشد.

در راستای نیل به اهداف مذکور، تدوین استانداردهایی به منظور متمرکز سازی و ارائه راهکارهای مناسب که تعامل میان سامانه‌های دولتی با درگاه خدمات الکترونیک ایران را امکانپذیر سازد، هدف اصلی این مستند می‌باشد. بدین منظور ابتدا نگاهی اجمالی به روشها، استانداردها و تکنولوژی‌های مطرح جهت یکپارچه‌سازی سامانه‌های نرم‌افزاری خواهیم داشت و سپس به انتخاب یکی از آنها به عنوان تکنولوژی برگزیده برای درگاه خدمات الکترونیک ایران می‌پردازیم. استانداردهای برگزیده باید دارای ویژگی‌هایی همچون همبستگی زیاد، وابستگی کم، استقلال از پلتفرم و مبتنی بر استانداردها و پروتکل‌های مطرح در دنیا باشد.

در خاتمه، خواهشمند است، پیشنهادات، انتقادات و نظرات کارشناسانه خود را جهت بهبود این سند به آدرس it@iran.ir ارسال فرمایید.

با تشکر

معاونت فناوری اطلاعات

وزارت ارتباطات و فناوری اطلاعات

سر آغاز

به منظور یکپارچه‌سازی سامانه‌های نرم‌افزاری معماری‌های متعددی وجود دارد که یکی از مطرح‌ترین آنها معماری سرویس‌گرا (SOA) می‌باشد.

معرفی معماری سرویس‌گرا

معماری سرویس‌گرا، شکل تکامل یافته ساختار توزیع شده، مبتنی بر فرضیه طراحی تقاضا/پاسخ در سامانه‌ها و برنامه‌های کاربردی همگام و ناهمگام است. منطق کلی سامانه سرویس‌دهنده به صورت ماژولار ارائه می‌شود و به عنوان سرویس‌هایی در اختیار برنامه‌های مصرف‌کننده (Client) قرار می‌گیرد. مهم‌ترین نکته این معماری طبیعت اتصال آزادانه مصرف‌کنندگان است؛ بدین معنی که سامانه‌های خدمت‌دهنده و خدمت‌گیرنده مستقل از پیاده‌سازی‌های گوناگون نرم‌افزاری هستند. توسعه‌دهندگان برنامه‌های خدمت‌گیرنده که معمولاً سرویس‌های مختلف را از سامانه‌های متعدد گردآوری و استفاده می‌کنند، در این معماری می‌توانند، بدون آگاهی از نحوه پیاده‌سازی زیرین سرویس‌دهنده‌ها، از سرویس‌های مورد نظر استفاده کنند.

برای مثال، یک سرویس می‌تواند تحت پلاتفرم .Net یا J2EE پیاده‌سازی گردد، و برنامه کاربردی استفاده‌کننده از این سرویس می‌تواند بر روی یک پلاتفرم دیگر قرار داشته و با استفاده از یک زبان برنامه‌سازی ثالث پیاده‌سازی شود.

مفهوم معماری SOA اساساً مفهوم جدیدی نیست، ولی نسبت به تکنولوژی‌های توزیع‌شده ماقبل خود تفاوتها و مزایایی دارد که اغلب شرکتهای مطرح دنیا آن را پذیرفته و دارای یک مجموعه ابزار یا کتابخانه‌های کاربردی هستند که توسعه برنامه‌های کاربردی را در این معماری امکان‌پذیر می‌سازد.

ویژگیهای معماری سرویس‌گرا

- اجزای نرم افزارها در SOA سرویس‌ها هستند که مبتنی بر پروتکل‌هایی استاندارد می‌باشند.
- سرویس‌ها در SOA دارای کمترین میزان وابستگی به یکدیگر می‌باشند.
- بستر ارتباطی استانداردهای SOA می‌بایست بگونه‌ای باشد که فاقد وابستگی به پروتکل‌های خاص و لایه‌های زیرین شبکه باشد.
- سرویس‌گیرندگان به دور از پیچیدگی محتوای سرویس‌ها، فقط با واسطه‌های آنها در ارتباط هستند.
- سرویس‌ها می‌توانند از پلاتفرم‌های متفاوت بوده و با یکدیگر تعامل داشته باشند.

تکنولوژی‌های یکپارچه سازی

پس از مروری بر معماری مطرح در مجتمع سازی برنامه‌های کاربردی ، به بررسی برخی تکنولوژی‌های برتر جهت یکپارچه سازی سامانه های نرم‌افزاری می پردازیم.

تکنولوژی پیغام رسانی (Messaging)

در این تکنولوژی ارتباط بین سامانه‌های مختلف از طریق ارسال و دریافت پیغام صورت می‌پذیرد به طوری‌که سامانه مبدأ هر آنچه را که از سامانه مقصد می‌خواهد در قالب یک پیغام به آن ارسال می‌کند و سامانه مقصد نیز اطلاعات مورد نیاز سامانه مبدأ را در قالب پیغام دیگری برای آن ارسال می‌نماید.

به طور کلی فرایند ارسال پیام به دو صورت نقطه به نقطه (peer to peer) و انتشاری (publish and subscribe) انجام می‌شود که از حالت اول زمانی که تبادل پیام بین دو سامانه صورت می‌گیرد و از حالت دوم زمانی که پیغام‌ها ما بین چندین سامانه رد و بدل می‌شوند استفاده می‌گردد.

با توجه به اینکه ارسال پیام به صورت غیر همزمان (asynchronous) بین سامانه‌های مختلف انجام می‌پذیرد در صورتیکه سامانه مقصد در لحظه دریافت پیغام به هر علتی (مثلا وجود نقص در سیستم ارتباطی و یا عدم توانایی در برقراری ارتباط با پایگاه داده خود) قادر به پاسخگویی به آن نباشد، اختلالی در عملکرد سیستم پیام رسان به وجود نمی‌آید چرا که با توجه به قرارگیری پیغام‌ها در یک صف (Message Queue) سامانه مبدأ مجبور به ارسال مجدد پیام نمی‌باشد و نیز ادامه کار سامانه پیام رسان منوط به دریافت پاسخ از طرف پیام گیرنده نیست.

در صورت تمایل به استفاده از این تکنولوژی جهت یکپارچه سازی برنامه های کاربردی - نظر به وجود API های متنوع برای پیاده سازی تکنولوژی مزبور و مقید شدن برنامه کاربردی به API ای که از آن استفاده می‌کند- لازم است آن برنامه ها، یا از یک API واحد استفاده نمایند و یا طوری طراحی شوند که توانایی کار با API های یکدیگر را داشته باشند.

یکی از معتبرترین و پرکاربردترین API ها به منظور پیاده سازی این تکنولوژی در صورت استفاده از پلاتفرم J2EE ، JMS ، (Java Messaging Service) و در صورت استفاده از پلاتفرم .NET ، Microsoft Message Queuing (MSMQ) می‌باشد.

تکنولوژی وب سرویس

یک وب سرویس، سرویسی است که از طریق وب قابل دسترسی بوده، از یک سامانه پیغام رسانی استاندارد مبتنی بر XML - با یک دستور زبان معین - استفاده نموده، وابستگی خاصی به هیچ پلاتفرمی - اعم از سیستم عامل و زبان برنامه نویسی - نداشته و نیز توسط یک مکانیزم جستجوی ساده قابل دسترسی باشد.

عدم وابستگی وب سرویسها به پلاتفرمی خاص ، تقید به رعایت استانداردهایی معین ، همچنین سهولت در بازیابی آنها از طریق وب ، آنها را تبدیل به نامزد مناسبی جهت یکپارچه سازی سامانه های نرم افزاری نموده است.

به طور کلی سه نقش اصلی در معماری وب سرویس‌ها قابل شناسایی است :

فراهم کننده سرویس (Service provider)، که سرویس را پیاده سازی نموده و امکان دسترسی به آن را از طریق بستر وب مهیا می‌سازد.

درخواست کننده سرویس (Service requester)، که همان استفاده کننده سرویس است و درخواست خود را در قالب فرمت XML برای فراهم کننده سرویس ارسال می‌کند.

محل ثبت سرویس (Service registry)، که محلی منطقاً متمرکز، جهت ثبت سرویسها می باشد، به طوریکه فراهم کننده سرویس می‌تواند سرویسهای خود را در آنجا منتشر کرده و درخواست کننده سرویس ، سرویس‌های مورد نظر خود را در آنجا بیابد.

از نقطه نظر معماری نیز مجموعه پروتکل‌های سرویسها شامل لایه های زیر می باشد:

لایه انتقال سرویس (Service transport)، که وظیفه انتقال پیغامها بین برنامه‌های کاربردی را بر عهده دارد و در حال حاضر شامل پروتکل‌های HTTP، FTP، SMTP و BEEP می‌باشد.

لایه پیام رسانی مبتنی بر XML (XML messaging)، که وظیفه آن کد کردن پیغامها در قالب فرمت XML استاندارد است که هم برای فراهم کننده سرویس قابل درک باشد و هم برای درخواست کننده آن.

لایه توصیف سرویس (Service description)، که یک واسط (رابط) عمومی را برای یک وب سرویس خاص تعریف می‌کند (WSDL).

لایه شناسایی سرویس (Service discovery)، که وظیفه آن متمرکز کردن سرویسها در یک محل ثبت مشترک است به گونه ای که فراهم کننده سرویس قادر به انتشار سرویس خود در آنجا بوده و درخواست کننده سرویس نیز بتواند سرویس مورد نظر خود را در آنجا بیابد. این لایه در حال حاضر توسط فناوری استاندارد UDDI به نام اداره می شود.

وب سرویس ها یکی از روشهای استاندارد پیاده سازی معماری سرویس گرا می باشند که توانایی تبادل اسناد ساخت یافته با حجم اطلاعات متفاوت را دارا بوده و قادر هستند مشخصات و ویژگیهای مربوط به اطلاعات موجود در اسناد- فرا داده‌ها - را مبادله کنند.

از جمله معتبرترین شرکتهای پیشرو در زمینه بسط و توسعه وب سرویس‌ها و نیز framework ها و ابزارهایی جهت پیاده سازی آنها عبارتند از: Microsoft، Sun و IBM که همه آنها در محصولات خود از استانداردهای تدوین شده برای این تکنولوژی تبعیت می‌کنند.

تکنولوژی CORBA

این تکنولوژی علاوه بر دارا بودن ویژگی های مطروحه در تکنولوژی وب سرویس - عدم وابستگی به پلتفرم و تبعیت از استانداردهایی معین - در برخی زمینه ها دارای برتری هایی نیز نسبت به آن می باشد.

تکنولوژی CORBA استاندارد جهت یکپارچه سازی سامانه های شی گرا بوده به طوریکه با فراهم نمودن گذرگاهی جهت تبادل اشیا (object ها) امکان فراخوانی متدهای موجود در آنها را به صورت remote، مستقل از نوع پلتفرمی که بر روی آن قرار دارند و نیز محل قرار گیری فیزیکی آنها فراهم می سازد.

عمده ترین وجه تمایز بین دو تکنولوژی CORBA و وب سرویس آن است که اولی مبتنی بر object بوده و بر خلاف مورد دوم که در آن ارسال اشیا به صورت پیام هایی در قالب ساختار XML صورت می گیرد اشیا به همان شکل اولیه بین سرویس دهنده و سرویس گیرنده رد و بدل می شوند .

همانگونه که وب سرویسها از پروتکل SOAP جهت تبادل پیام استفاده می کنند در CORBA نیز به منظور تبادل اشیا از پروتکل IIOP استفاده می گردد، همچنین زبان تعریف واسط در این تکنولوژی IDL بوده و تعامل ما بین سرویس دهنده و سرویس گیرنده به واسطه ORB(object request brokers) انجام می شود.

همانگونه که پیشتر ذکر شد استفاده از تکنولوژی CORBA نسبت به وب سرویس ها مزایایی دارد که از آن جمله می توان به سرعت بالاتر آن در ارسال پیامها ، نیاز به پهنای باند کمتر و نیز سهولت بیشتر در تجزیه و تحلیل پیامهای ارسالی بر روی پروتکل IIOP اشاره نمود.

تکنولوژی گذرگاه مشترک سرویس (ESB)

یکی از تکنولوژیها به منظور یکپارچه سازی برنامه های کاربردی استفاده از یک گذرگاه مشترک جهت مدیریت و اداره کردن پیام های ارسال شده و نیز پیام های دریافتی می باشد. در این شیوه هر سامانه اطلاعات مورد نیاز خود را در قالب یک پیام به گذرگاه ارسال کرده و گذرگاه این اطلاعات را به سامانه مقصد تحویل می دهد به عبارت دیگر مسوولیت مدیریت و اداره کردن پیام های ارسالی از سامانه مبدا به سامانه مقصد و بالعکس بر عهده گذرگاه مشترک خواهد بود. در نتیجه اداره کردن پیچیدگی های مربوط به استفاده از تکنولوژی پیام رسانی (مثلا JMS در صورت استفاده از پلاتفرم J2EE) - که در صورت عدم به کارگیری ESB توسط برنامه نویس انجام می شود - به عهده گذرگاه مشترک گذاشته می شود.

از دیگر مزایای استفاده از این تکنولوژی امکان یکپارچه سازی سامانه های مختلف با پلاتفرمها و معماری های متفاوت - مثلا

SOA (Service Oriented Architecture) و EDA (Event Driven Architecture) و نیز legacy system ها- با یکدیگر بدون نیاز به اعمال تغییرات گسترده در برنامه های کاربردی می باشد به طوری که می توان برنامه های کاربردی تحت وب مبتنی بر پلاتفرمهای .NET و j2ee و غیره را با یکدیگر و حتی با برنامه های کاربردی مبتنی بر پلاتفرمهای قدیمی که تحت وب نیستند (legacy systems) - که تعدادشان اندک هم نیست - مجتمع سازی نمود.

همچنین استفاده از این تکنولوژی متضمن اطمینان پذیری بالا در تبادل پیامها بین فرستنده و گیرنده می باشد چرا که در صورت بروز هرگونه مشکلی که منجر به عدم ارسال صحیح پیام از مبدا به مقصد شود گذرگاه مشترک تا حصول اطمینان از دریافت صحیح پیام توسط مقصد آن را مجددا ارسال خواهد نمود که این موضوع خود باعث ساده تر شدن فرایند مجتمع سازی و عدم نیاز به درگیر شدن با مسائل مربوط به ارسال پیام می گردد .

پیاده سازی های متنوعی برای تکنولوژی مذکور وجود دارد که دو مورد از آنها که مبتنی بر پلاتفرم J2EE بوده و به صورت رایگان عرضه شده و بیشترین تطابق را با معیار (استاندارد) های یک ESB دارند عبارتند از : Apache ServiceMix و Mule.

برای مثال Neuron ESB و ESB.NET نمونه هایی از ESB های پیاده سازی شده در محیط .NET می باشند.

علاوه بر تکنولوژی‌های فوق‌الذکر تکنیک‌های دیگری نیز به منظور یکپارچه‌سازی سامانه‌های نرم‌افزاری وجود دارد که بررسی و بحث در مورد آنها خارج از حوصله این مستند است، لیکن از اهم آنها می‌توان به موارد ذیل اشاره نمود:

COM+ ، JNBridge، Windows Communication Foundation (WCF)، .NET Remoting

،DCOM و غیره.

نتیجه گیری

با توجه به گستردگی استفاده از سرویس‌های وب و همچنین وجود استانداردهایی به منظور طراحی و پیاده سازی آنها و نیز تعدد پلاتفرم‌ها و نرم افزارهایی که از آنها پشتیبانی می‌کنند بعلاوه وجود ابزارهای متعدد در پلاتفرم‌های مختلف (اعم از J2EE ، NET و غیره) که امر طراحی و پیاده سازی سرویس‌های وب را تسهیل می نمایند، اینگونه به نظر می‌رسد که در حال حاضر استفاده از آنها به منظور مجتمع سازی برنامه‌های کاربردی بهترین راه حل باشد، لیکن به کارگیری تکنولوژی CORBA، با توجه به مزایای قابل تاملی که نسبت به وب سرویسها دارد، در آینده، امری دور از ذهن به نظر نمی‌رسد.

درگاه خدمات دولت الکترونیک ایران بعنوان مدخلی برای ارائه و یا دسترسی به سرویسها و خدمات موجود در سازمانهای دولتی و نهادهای وابسته به آن، با استفاده از استاندارد وب سرویسها، توانایی برقراری ارتباط و اتصال به سرویس دهنده‌ها و سیستم‌های توزیع شده موجود در سازمانها را دارد. در راستای تحقق اهداف دولت الکترونیک، دستگاه‌های اجرایی، سازمانها و نهادهای تابعه می باید با ارائه خدمات الکترونیکی خود و تحلیل و برآورد نیازهای سخت افزاری و نرم افزاری، سعی در ایجاد بستری جهت ارائه سرویس‌های مبتنی بر وب مطابق با استانداردها نمایند. در این راستا انتخاب پلاتفرم پیاده سازی وب سرویسها در حیطه اختیارات سرویس دهندگان (سازمانها) می باشد که بهتر است از محصولات و تکنولوژی‌های نرم افزاری مجوز دار، مانند محصولات و برنامه های آزاد/متن باز استفاده شود. پس از پیاده سازی سرویسها، جهت به کارگیری موثر آنها در درگاه خدمات الکترونیک ایران، مقتضی است مستندات دسترسی و استفاده از آنها در اختیار درگاه ایران قرار گیرد.

Marina Fisher, Ray Lai, Sonu Sharma, Laurence Moroney. (2006). Java EE and .Net Interoperability: Integration Strategies, Patterns, and Best Practices. Prentice Hall.

Ethan Cerami.(2002). *WebServices Essentials* (first Ed.). O'Reilly.

Inderjeet Singh, Sean Brydon, Greg Murray, Vijay Ramachandran, Thierry Violleau & Beth Stearns. (2004). *Designing Web Services with the J2EE™ 1.4 Platform JAX-RPC, SOAP, and XML Technologies*. Addison Wesley.

Tijs Rademakers & Jos Dirksen.(2007). Understanding ESB functionality: The world of Open-Source ESBs. *Open-Source ESBs in Action*. Manning Publications.

<http://www.ibm.com/developerworks/xml/library/ws-esbscen/>

<http://incubator.apache.org/servicemix/home.html>

<http://www.ddj.com/java/201200303>