



وب سرویس ها

مفاهیم پایه وب سرویس ها ۳

حسن دولتی

XML (EXTENSIBLE MARKUP LANGUAGE)

XML یک تکنولوژی است که به شکل گسترده از آن پشتیبانی می شود ، همچنین این تکنولوژی Open است به این معنی که تعلق به شرکت خاصی ندارد . اولین بار در کنسرسیوم WWW یا W3C در سال ۱۹۹۶ برای ساده کردن انتقال دیتا ایجاد شده است.

با گسترده شدن استفاده از وب در دهه ۹۰ کم کم محدودیت های HTML مشخص شد . ضعف HTML در توسعه پذیری (قابلیت اضافه و کم کردن خواص) و ضعف آن در توصیف دیتاهایی که درون خود نگهداری می کند برنامه نویسان را از آن نا امید کرد . همچنین مبهم بودن تعاریف آن باعث شد از توسعه یافتن باز بماند .

در پاسخ به این اشکالات W3C یک سری امکانات را در جهت توسعه HTML به آن افزود که امکان تغییر ساختار متن های HTML مهم ترین آن است . این امکان را CSS یا Cascade Style Sheet می نامند . این توسعه تنها یک راه موقتی بود. باید یک روش استاندارد شده ، توسعه پذیر و دارای ساختار قوی ایجاد می شد .

در نتیجه W3C ، XML را ساخت. XML دارای قدرت و توسعه پذیری SGML یا Standard Generalized Markup Language و سادگی که در ارتباط در وب به آن نیاز دارد است .

XML-ادامه

استقلال اطلاعات یا جدا بودن محتوا از ظاهر یک مشخصه برای XML به حساب می آید . متن های XML فقط یک دیتا را توصیف می کنند و برنامه ای که XML برای آن قابل درک است بدون توجه به زبان و سیستم عامل - قادر است به اطلاعات درون فایل XML هر گونه شکلی که مایل است بدهد . متن های XML حاوی دیتا هستند بدون شکل خاص بنابراین برنامه ای که از آن می خواهد استفاده کند باید بداند که چگونه می خواهد آن اطلاعات را نمایش دهد .

بنابر این نحوه نمایش یک فایل XML در یک PC با PDA و تلفن همراه می تواند متفاوت باشد .

وقتی یک برنامه با متن XML مواجه می شود باید مطمئن باشد که آن متن حاوی دیتای مورد نظر خود است . این اطمینان توسط برنامه هایی با نام XML Parser حاصل می شود . تجزیه کننده ها دستورات متن XML را بررسی می کنند . همچنین آنها به برنامه کمک می کنند تا متن های XML را تفسیر کند .

XML-ادامه

به صورت اختیاری هر متن XML می تواند به متن دیگری اشاره کند که حاوی ساختار فایل XML اصلی باشد . به آن متن XML دوم DTD یا Document Type Definition گفته می شود . وقتی فایل XML به DTD اشاره می کند برنامه تجزیه کننده فایل اصلی را با DTD بررسی می کند که آیا به همان ساختاری که در DTD توصیف شده شکل گرفته است یا خیر . اگر یک تجزیه کننده XML بتواند یک متن را به درستی پردازش کند متن XML نیز به شکل صحیحی فرمت شده است .

وقتی که اکثر نرم افزار ها امکانات وبی خود را افزایش دادند این طور به نظر می آید که XML به عنوان یک تکنولوژی جهانی برای فرستادن اطلاعات بین برنامه های انتخاب شود .

تمامی برنامه هایی که از XML استفاده می کنند قادر خواهند بود که XML همدیگر را بفهمند . این سطح بالای تطابق بین برنامه ها باعث می شود که XML یک تکنولوژی مناسب برای وب سرویس باشد ، چون بدون اینکه احتیاج به سیستم عامل و سخت افزار یکسان باشد می تواند اطلاعات را جابجا کند .

LANGUAGE WEB SERVICES DESCRIPTION : (WSDL)

✓ استاندارد دیگری که نقش اساسی در وب سرویس بازی می کند WSDL است . همانطور که قبلا اشاره کردیم یکی از خواص وب سرویس ها توصیف خود آنها است به این معنی که وب سرویس دارای اطلاعاتی است که نحوه استفاده از آن را توضیح می دهد . این توضیحات در WSDL نوشته می شود ، متنی در قالب XML که به برنامه ها می گوید این وب سرویس چه اطلاعاتی لازم دارد و چه اطلاعاتی را بر می گرداند .

✓ وقتی که سازندگان نرم افزار برای اولین بار SOAP و دیگر تکنولوژی های وب سرویس را ساختند دریافتند که برنامه ها قبل از اینکه شروع به استفاده از یک وب سرویس بکنند باید اطلاعاتی درباره آن را داشته باشند . اما هر کدام از آن سازندگان برای خودشان روشی برای ایجاد این توضیحات ابداع کردند و باعث شد که وب سرویس ها با هم هماهنگ نباشد . وقتی IBM و مایکروسافت تصمیم گرفتند تا استانداردهای خود را یکسان کنند WSDL بوجود آمد .

✓ در ماه مارس سال ۲۰۰۱ مایکروسافت ، IBM و Ariba نسخه ۱.۱ را به W3C ارائه کردند . گروهی از W3C بر روی این استاندارد کار کردند و آن را پذیرفتند .

WSDL - ادامه

- هر وب سرویسی که بر روی اینترنت قرار می گیرد دارای یک فایل WSDL است که مشخصات ، مکان و نحوه استفاده از وب سرویس را توضیح می دهد .
- مستند WSDL متنی است در قالب XML حاوی اطلاعاتی بیانگر اینکه وب سرویس به چه اطلاعاتی احتیاج داشته و چه اطلاعاتی را باز میگرداند .
- مستند WSDL شامل اطلاعات مکان و نحوه استفاده از وب سرویس از قبیل پارامترهای ورودی و خروجی هر وب سرویس میباشد . کاربر احتیاجی به دانستن ساختار و دستورات WSDL نخواهد داشت چرا که اکثر نرم افزارهای طراحی وب سرویس بصورت خودکار مستند WSDL لازم را نیز تولید می نمایند .
- یک فایل WSDL نوع پیغام هایی که وب سرویس می فرستد و می گیرد را توضیح می دهد مانند پارامترهایی که برنامه صدا زننده برای کار با وب سرویس باید به آن بفرستد . در تئوری یک برنامه در وب برای یافتن وب سرویس مورد نظر خود از روی توضیحات WSDL ها جستجو می کند . در WSDL اطلاعات مربوط به چگونگی ارتباط با وب سرویس بر روی HTTP یا هر پروتکول دیگر نیز وجود دارد .

WSDL - ادامه

این مهم است که بدانیم WSDL برای برنامه ها طراحی شده است نه برای خواندن آن توسط انسان .
شکل فایل های WSDL پیچیده به نظر می آید ولی کامپیوترها می توانند آن را بخوانند و تجزیه و تحلیل
بکند .

خیلی از نرم افزارهایی که وب سرویس می سازند فایل WSDL مورد نیاز وب سرویس را نیز تولید می
کنند بنابراین وقتی برنامه نویس وب سرویس خود را ساخت به شکل خودکار WSDL مورد نیاز با آن
نیز ساخته می شود و احتیاجی به آموزش دستورات WSDL برای ساختن و استفاده از وب
سرویس نیست.

انتشار سرویس وب (UDDI)

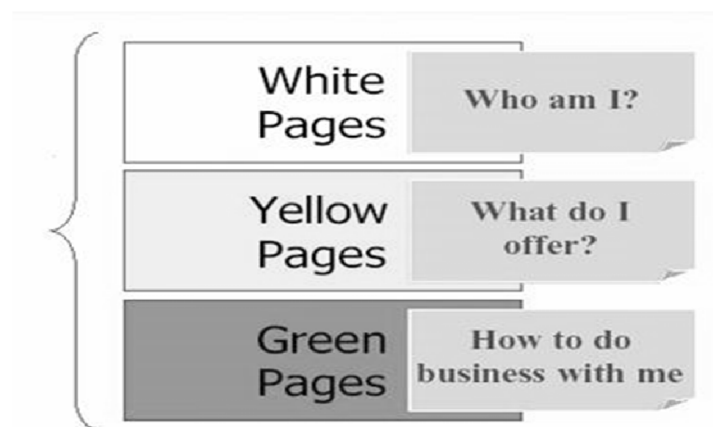
○ «کشف و یکپارچه سازی جهانی توضیحات» (Universal Description Discovery and Integration)

○ طراحی برای یکپارچه سازی و هماهنگ کردن سرویس های وب مستقل در سراسر دنیا است . هدف UDDI یکسان کردن عملیات کشف، توضیح و اجرای سرویس های وب (براساس نیازهای تجاری) است.

○ با استفاده از UDDI شرکت ها می توانند اطلاعات خود را در اختیار شرکت های دیگر قرار بدهند و مدل B ۲ ایجاد کنند . همان طور که از نام آن مشخص است شرکت ها می توانند وب سرویس خود را معرفی کنند ، با وب سرویس دیگران آشنا شوند و از آن در سیستم های خود استفاده کنند. این استاندارد در سال ۲۰۰۰ ساخته شده ، کنسرسیومی از شرکت های صنعتی در حال کار بر روی آن هستند ؛ نسخه دوم UDDI در ماه ژوئن سال ۲۰۰۱ ارائه شد.

UDDI

○ پرس و جو از این پایگاه با استفاده از پروتکل SOAP انجام می پذیرد. اطلاعات درون پایگاه UDDI به سه دسته صفحات سفید (که شامل اطلاعات شرکت ها و توضیحات متنی درباره آنها می باشند) ، صفحات زرد (شامل اطلاعات توانایی های فنی شرکتها در گروه بندی های مختلف می باشند) و صفحات سبز (که شامل اطلاعات فنی و تکنیکی در مورد وب سرویس ها می باشند) تقسیم میگردند .



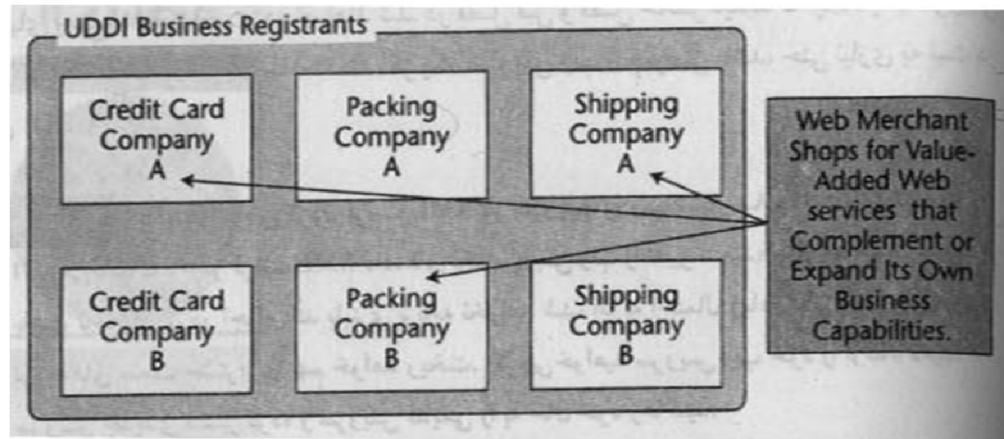
UDDI-ادامه

✓ ساختار UDDI شبیه یک کتاب بزرگ است: در فهرست سایت **UDDI.org** لیستی از تمام مراکز ثبت کننده موجود (که به میزبان UDDI معروفند) وجود دارد. درست مثل یک کتاب با خواندن فهرست **UDDI.org** نمی توان از محتویات سایت های میزبان UDDI آگاهی یافت.

✓ با اینکه می توان برای یافتن یک سرویس وب مستقیماً به سایت میزبان آن رفت، ولی این کاری بس دشوار است. برای این منظور می توان از امکانات درونی ابزارهایی مانند چارچوب **.NET**. (با امکانات جستجو در سایت **UDDI.org**) استفاده کرد.

✓ علاوه بر دسترسی به ثبت کننده های UDDI در سایت **UDDI.org** می توانید در این سایت خود را بعنوان یک «شریک UDDI» نیز ثبت کنید. شریک UDDI یعنی شما خود میزبان سرویس وب خودتان هستید- در حالیکه نیازی نیست که یک ثبت کننده باشید.

✓ از ویرایش اول UDDI فقط میکروسافت و IBM پشتیبانی می کنند، در حالیکه هر چهار شرکت فوق (میکروسافت، IBM، هیولت پاکارد و SAP) از ویرایش دوم آن پشتیبانی می کنند. برای کسب اطلاعات بیشتر درباره ویرایش دوم UDDI به قسمت **Technical Highlights** در صفحه **www.Uddi.org** مراجعه کنید.



در این شکل شش شرکت را می بینید که سرویس هایی در سه زمینه مختلف ارائه می کنند. دو تای آنها کارت اعتباری اینترنتی می فروشند، دو تای دیگر خدمات بسته بندی عرضه می کنند، و دو تای آخر در زمینه حمل و نقل فعالیت می کنند. اگر روزی تصمیم گرفتید از سرویس های این شرکتها استفاده کنید، می توانید تصمیم گیری خود را براساس اطلاعاتی که به همراه سرویس های وب آنها هست، قرار دهید. اما ، پیدا کردن جای این سرویس ها برعهده میزبان UDDI است .

UDDI های عمومی و خصوصی

- تا اینجا درباره سرویس های وب از جنبه عمومی صحبت کردیم، که هر کاربری با دسترسی صحیح UDDI می تواند سرویس را کشف و از آن استفاده کند. البته به کمک .NET و با استفاده از فایل Web.Config می توان دسترسی یک سرویس را به کاربران خاص محدود کرده و آنها را مجبور به احراز هویت نمود. با این حال، کاربران در هر صورت می توانند این سرویس را روی UDDI «ببینند».
- تا قبل از UDDI 2.0 اگر نمی خواستید یک سرویس وب دیده شود، باید از انتشار آن روی میزبانهای UDDI عمومی بکلی صرفنظر می کردید. اما UDDI 2.0 اجازه می دهد که سرویس های وب خصوصی را روی یک میزبان UDDI عمومی قرار دهید.
- یکی دیگر از آیتم های جدید در UDDI 2.0 «زبان بازدید سرویس وب» (WSIL) است، که از سوی میکروسافت و IBM معرفی شده است. در واقع WSIL یک ابزار کمکی برای UDDI در جستجوی ساده تر سرویس های وب است. بعبارت دیگر، اگر URL یک سرویس وب را بدانید دیگر نیازی نیست برای استفاده از آن سراغ میزبان های UDDI بروید. وقتی بدانید که هر روز هزاران سرویس وب جدید به UDDI اضافه می شود، ارزش چنین میانبری را بهتر درک خواهید کرد.

SIMPLE OBJECT ACCESS PROTOCOL (SOAP)

□ پروتکل دسترسی شیء ساده (یا همان SOAP) در سال ۱۹۹۸ توسط میکروسافت (همزمان با کیت ابزار Microsoft SOAP 1.0) معرفی شد. هدف از ارائه این کیت دستیابی به تکنیکی برای ارسال و دریافت داده ها روی اینترنت با استفاده از انواع داده تعریف شده و بین ماشین های مختلف بود.

□ این پروتکل حاصل تلاش مشترک شرکت هایی از قبیل میکروسافت، IBM، Lotus Development، Userland Software و DevelopMentor بود، که توسط W3C (کنسرسیوم شبکه تارنمای جهانی) نیز به عنوان یک استاندارد به رسمیت شناخته شده و تقویت شد. برای کسب اطلاعات بیشتر می توانید به www.w3c.org/TR/SOAP مراجعه کنید.

SOAP- مرور پروتکل ها

■ پروتکل کنترل انتقال/پروتکل اینترنت، یا همان TCP/IP معروف، سردمدار پروتکل هائی است که به پروتکل های TCP معروف هستند، و از میان آنها می توان به HTTP، DHCP، SSH، SNMP، NNTP، POP3، SMTP، NFS، IMAP، RSH، FTP، DNS، Rlogin، Telnet، SOAP، اکثر پروتکل های گپ اینترنتی و پروتکل های X Windows اشاره کرد. از میان اینها سه پروتکل HTTP، FTP و SMTP بیشترین کاربرد را در دنیای اینترنت دارند. اینها پروتکل های متنی هستند، که می توان بصورت خودکار (در کد برنامه) و یا تعاملی (نشست زنده کاربر) از آنها بهره گرفت.

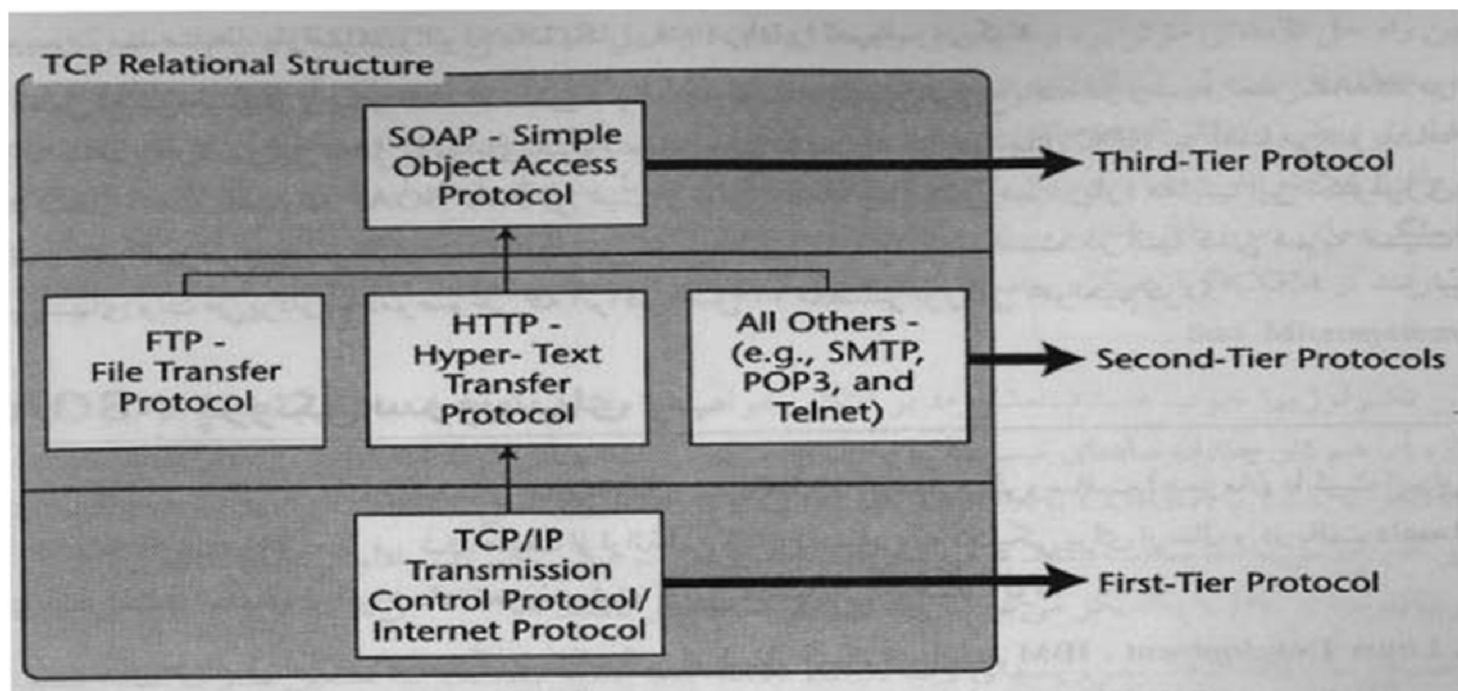
■ از این میان پروتکل HTTP بیشترین سهم را در تبادل اطلاعات معنادار و ایجاد سیستم های قدرتمند مشتری/ سرویس دهنده بازی می کند. برای مثال سرویس دهنده IIS می تواند به بهترین وجه درخواست های HTTP (حتی درخواستهایی که شامل کد هستند مانند صفحات ASP) را پاسخ دهد. به دلیل ارتباط تنگاتنگ IIS با MTS (سرویس دهنده تراکنش) و COM+ (که به سرویس های کامپوننت معروفند) مشتری می تواند از منابع سرویس دهنده (مانند سرویس های پایگاه داده) بهره بگیرد.

SOAP-ادامه

□ در واقع هدف از ارائه پروتکل SOAP غلبه بر مشکلات موجود بر سر راه ایجاد سرویس های TCP بود. از آنجائی که پروتکل SOAP اساساً یک پروتکل TCP است و از XML استفاده می کند، می تواند خود را در قالب درخواست/پاسخ های HTTP مخفی کرده، براحتی از هر نوع فایروالی عبور کند. در حقیقت SOAP نیز (مانند HTTP) یک پروتکل TCP است، و بدون آن نمی تواند وجود داشته باشد. به بیان دیگر، SOAP را می توان لایه سوم روی لایه های HTTP و TCP (که ارتباط فیزیکی را فراهم می آورد) فرض کرد (به شکل صفحه بعد نگاه کنید).

□ مذاکره بر سر مشخصات ارتباط و فراخوانی روال از راه دور (RPC) برعهده پروتکل TCP/IP است. در طی این مذاکره معلوم می شود که دو برنامه چگونه می توانند با هم ارتباط برقرار کنند. RPC در واقع راهی است که یک کامپیوتر کدی را روی کامپیوتر دیگر اجرا می کند.

SOAP - ادامه



SOAP - ادامه

اگر یک سرویس دهنده تبادل اطلاعات هر دو جهت ارسال و دریافت را بپذیرد (مانند TCP/IP)، به آن ارتباط دوجانبه یا متقابل می گویند. اما اگر سرویس دهنده چیزی بیش از یک گیرنده اطلاعات نباشد و هیچ نوع اطلاعاتی (حتی موفقیت یا عدم موفقیت پردازش خواسته شده) را برنگرداند (مانند پروتکل UDP) به آن ارتباط یک جانبه (یا انگلی) گفته می شود. در سرویس های وب XML تقریباً همیشه ارتباط دوجانبه داریم.

SOAP پروتکلی از رده اول یا دوم نیست چون برای تفسیر اطلاعات ردوبدل شده به پروتکلی از رده دوم - در اینجا HTTP-متکی است. HTTP نیز بنوبه خود برای انتقال اطلاعات به TCP وابسته است. در واقع SOAP در رأس هرم پروتکل ها قرار می گیرد، ولی در عین حال دارای مشخصات استاندارد خاص خود نیز هست. آنچه SOAP را واقعاً قدرتمند و شایسته توجه کرده، توانایی آن در پشتیبانی از زبان ها و پلاتفرم های مختلف (از طریق XML) می باشد.

SOAP - ادامه

هدف اصلی SOAP ایجاد روش برای فرستادن دیتا بین سیستم هایی است که بر روی شبکه پخش شده اند . وقتی یک برنامه شروع به ارتباط با وب سرویس می کند ، پیغام های SOAP وسیله ای برای ارتباط و انتقال دیتا بین آن دو هستند .

یک پیغام SOAP به وب سرویس فرستاده می شود و یک تابع یا ساب روتین را در آن به اجرا در می آورد به این معنی که این پیغام از وب سرویس تقاضای انجام کاری می کند. وب سرویس نیز از محتوای پیغام SOAP استفاده کرده و عملیات خود را آغاز می کند. در انتها نیز نتایج را با یک پیغام SOAP دیگر به برنامه اصلی می فرستد.

به عنوان یک پروتکل مبتنی بر XML ، SOAP تشکیل شده از یک سری الگوهای XML ی است . این الگوها شکل پیغام های XML را که بر روی شبکه منتقل می شود را مشخص می کند ، مانند نوع دیتاها و اطلاعاتی که برای طرف مقابل تفسیر کردن متن را آسان کند .

اجزا پیغام SOAP

❖ در اصل SOAP برای انتقال دیتا بر روی اینترنت و از طریق پروتکل HTTP طراحی شده است ولی از آن در دیگر مدلها مانند LAN نیز می توان استفاده کرد . وقتی که وب سرویس ها از HTTP استفاده می کنند به راحتی می توانند از Firewall عبور کنند .

❖ یک پیغام SOAP از سه بخش مهم تشکیل شده است : پوشش یا Envelope برای بسته بندی کردن کل پیغام به کار می رود . این بخش محتوای پیغام را توصیف و گیرنده آن را مشخص می کند . بخش بعدی پیغام های SOAP ، Header آن است که یک بخش اختیاری می باشد و مطالبی مانند امنیت و مسیریابی را توضیح می دهد .

❖ بدنه یا Body پیغام SOAP بخشی است که دیتاهای مورد نظر در آن جای می گیرند . دیتاها بر مبنای XML هستند و از یک مدل خاص که الگوها (Schemas) آن را توضیح می دهند تبعیت می کنند . این الگوها به گیرنده کمک می کنند تا متن را به درستی تفسیر کند . پیغام های SOAP توسط سرور های SOAP گرفته و تفسیر می شود تا در نتیجه آن ، وب سرویس ها فعال شوند و کار خود را انجام دهند .

SOAP - ادامه

■ برای اینکه از SOAP در وب سرویس استفاده نکنیم از تعداد زیادی پروتکل باید استفاده شود. برای مثال XML-RPC تکنولوژی قدیمی تری بود که همین امکانات را ایجاد می کرد. به هر حال ، خیلی از سازندگان بزرگ نرم افزار SOAP را بر تکنولوژی های دیگر ترجیح دادند. دلایل زیادی برای انتخاب SOAP وجود دارد که خیلی از آنها درباره پروتکل آن است که فراتر از این متن می باشد. ۳ برتری مهم SOAP نسبت به تکنولوژی های دیگر، **Simplicity** : **Interoperability** و **Extensibility** است.

■ پیغام های SOAP معمولاً کدهای زیادی ندارند و برای فرستادن و گرفتن آن به نرم افزارهای پیچیده نیاز نیست. SOAP این امکان را به برنامه نویس می دهد تا بنا به نیاز خود آن را تغییر دهد. در آخر بدلیل اینکه SOAP از XML استفاده می کند می تواند بوسیله HTTP اطلاعات را انتقال بدهد بدون اینکه زبان برنامه نویسی ، سیستم عامل و سخت افزار برای آن مهم باشد.

کیت ابزار MICROSOFT SOAP

وقتی کیت ابزار Microsoft SOAP برای اولین در سال ۱۹۹۸ معرفی شد، هدف اصلی آن ارائه قابلیت ایجاد سیستم هایی مستقل از پلاتفرم با کمترین زحمت و با انتقال حداقل اطلاعات بود. قبل از SOAP ، فراخوانی هایی مانند Response.RedirectAction و Server.Transfer در صفحات ASP باعث می شد تا کل صفحه برای پردازش مجدد IIS فرستاده شود. البته اگر هدف ما رفتن به صفحه جدیدی باشد، این موضوع چندان مشکل ساز نیست. اما اگر فقط بخواهیم فیلدی کوچک در یک جدول بزرگ را تغییر دهیم، چنین رفتاری (پردازش کل صفحه در IIS) وقت گیر و پرهزینه خواهد بود.

کیت ابزار MICROSOFT SOAP

❖ در ویرایش دوم کیت ابزار SOAP قابلیت های آن باز هم بهبود یافته است. در این کیت روتین هایی وجود دارد که می توان با آنها فراخوانی هایی روی اینترنت انجام داد، فایل های wsml و wsdl را بطور خودکار ایجاد کرد، و انتقال اطلاعات را مدیریت کرد. اگر برنامه نویس ویژوال بیسیک ۶ هستید و می خواهید برنامه های خود را به سرویس وب تبدیل کنید، تنها راه SOAP Toolkit 2.0 است.

❖ SOAP یکی از عناصر ذاتی و تشکیل دهنده .NET Framework است، و نوشتن سرویس های وب با ویژوال استودیو .NET بسیار ساده و راحت شده است. با نصب .NET Framework بلافاصله آماده اید تا ایجاد سرویس های وب SOAP را شروع کنید.

.NET FRAMEWORK

- ❑ بسته نرم افزاری .NET Redistributable Framework. تمام چیزهایی را که برای نوشتن برنامه های .NET لازم است (ASP.NET, .NET Framework Class Library, CLR) در خود دارد. این برنامه فقط 21MB حجم دارد، ولی بایستی ملزومات نرم افزاری، سخت افزاری و نصب آنرا رعایت کنید.
- ❑ غیر از میکروسافت شرکت های دیگری نیز پشتیبانی از SOAP را به سیستم های خود اضافه کرده اند و از آنجائیکه SOAP اینک یک استاندارد جهانی محسوب می شود می توان انتظار داشت که در آینده تعداد آنها باز هم افزایش یابد.

مزایا و معایب SOAP

✓ قبل از .NET. برای استفاده از اشیاء و سرویس های SOAP هر مشتری باید این اشیاء را در کامپیوتر خود نصب می کرد. با آن که این اشیاء جای زیادی روی کامپیوتر نمی گرفتند، ولی پیکربندی آنها به کمی دقت نیاز داشت که برای اغلب کاربران معمولی ساده نبود. از طرف دیگر، کاربرانی که کامپیوترهای قدیمی داشتند نیز نمی توانستند بخوبی از سرویس های سمت-مشتری SOAP استفاده کنند، چون اجرای این کدها می توانست کامپیوتر را بشدت کند کرده و یا بکلی از کار بیندازد.

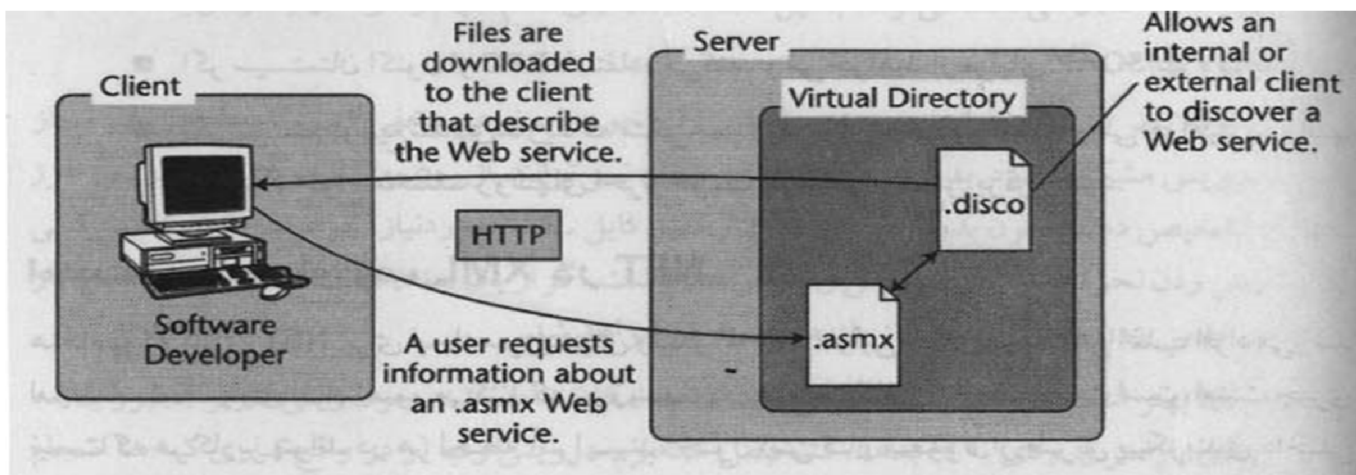
✓ علاوه بر آن، انواع داده ای که سرویس های اینترنتی SOAP پشتیبانی می کردند محدود بود. اگر برنامه نویسی می خواست از انواع داده خاصی استفاده کند، بایستی تغییراتی در SOAP می داد و این انحراف از استاندارد محسوب می شد.

مزایای SOAP

- ❖ SOAP پروتکلی سبک و جمع و جور است که اجازه می دهد فقط آن اطلاعاتی که واقعاً لازم است (نه کل صفحه وب) به سمت سرویس دهنده فرستاده شود. در صورت لزوم، SOAP حتی می تواند پردازش های سمت-سرویس گیرنده نیز انجام دهد.
- ❖ SOAP برخلاف سیستم های ارتباطی قبلی بر راحتی با سیستم مشتری یا سرویس گیرنده یکپارچه می شود؛ و با .NET. این زحمت حتی کمتر هم شده است.
- ❖ SOAP بصورت لایه ای روی HTTP کار می کند، و با کمترین زحمت و بدون نیاز به دستکاری در سیستم های مشتری و سرویس دهنده عمل می کند.
- ❖ SOAP یکی از استانداردهای توصیه شده W3C است.
- ❖ SOAP از سوی معتبرترین شرکت های نرم افزاری به رسمیت شناخته شده و پشتیبانی می شود.

مزایای SOAP-ادامه

- ❖ **SOAP** مشکل عبور اطلاعات از فایروال ها را (در سرویس های سمت-سرویس دهنده) حل کرده است.
- ❖ پشتیبانی از **SOAP** در **.NET**. تقریباً شفاف است. استفاده از سرویس های **SOAP** تفاوتی با فراخوانی توابع معمولی ندارد.
- ❖ **SOAP** با **.NET**. یکپارچه شده و نیازی به نصب نرم افزار جداگانه ندارد. فراخوانی سرویس های وب به همان سادگی فراخوانی تابع از **DLL** های محلی است.



معایب SOAP

- استانداردهای W3C در زمینه SOAP (مانند سایر استانداردهای W3C) صرفاً توصیه‌هایی بیش نیستند و بهمین دلیل شرکت‌های مختلف می‌توانند از استانداردهای داخلی خاص خود پیروی کرده و همه چیز را خراب کنند. با این اتفاق چیزی که SOAP با آن همه هیاهو (و امیدواری) به دنبال آن است، عملاً محقق نخواهد شد.
- SOAP همچنان از فقدان تنوع در انواع داده رنج می‌برد، چون HTTP قادر نیست تعداد نامحدودی از انواع داده را منتقل کند. البته در اغلب موارد تعداد داده مورد نیاز چندان زیاد نیست، ولی اگر برای یک برنامه خاص به UDP (نوع داده تعریف شده کاربر) نیاز دارید، باید قید SOAP را بزنید.
- با SOAP تقریباً هر کاری می‌شود کرد. اگر در یک سرویس وب تابعی برای فرمت کردن درایو C: قرار داده باشید، مطمئن باشید دیر یا زود کسی پیدا می‌شود که این کار را برایتان خواهد کرد.
- توجه کنترل دسترسی افراد به سرویس‌های وب اهمیت فوق‌العاده زیادی دارد، بویژه اگر می‌خواهید برنامه‌های **Active Directory** خود را بصورت سرویس وب درآورید.