

پردازش درخواست‌های HTTP در IIS

بگذارید در این قسمت خلاصه‌ای از درخواست‌های نوع HTTP را که تا به الان گفته‌ایم، به همراه شکل بیان کنیم: موقعی که کلاینت درخواست خود را مبنی بر یکی از منابع سرور ارسال می‌کند، Http.sys این درخواست را می‌گیرد. Http.sys با WAS تماس گرفته و درخواست می‌کند تا اطلاعات پیکربندی یا تنظیمات IIS را برای نحوه‌ی برخورد با درخواست، برایش بفرستد.

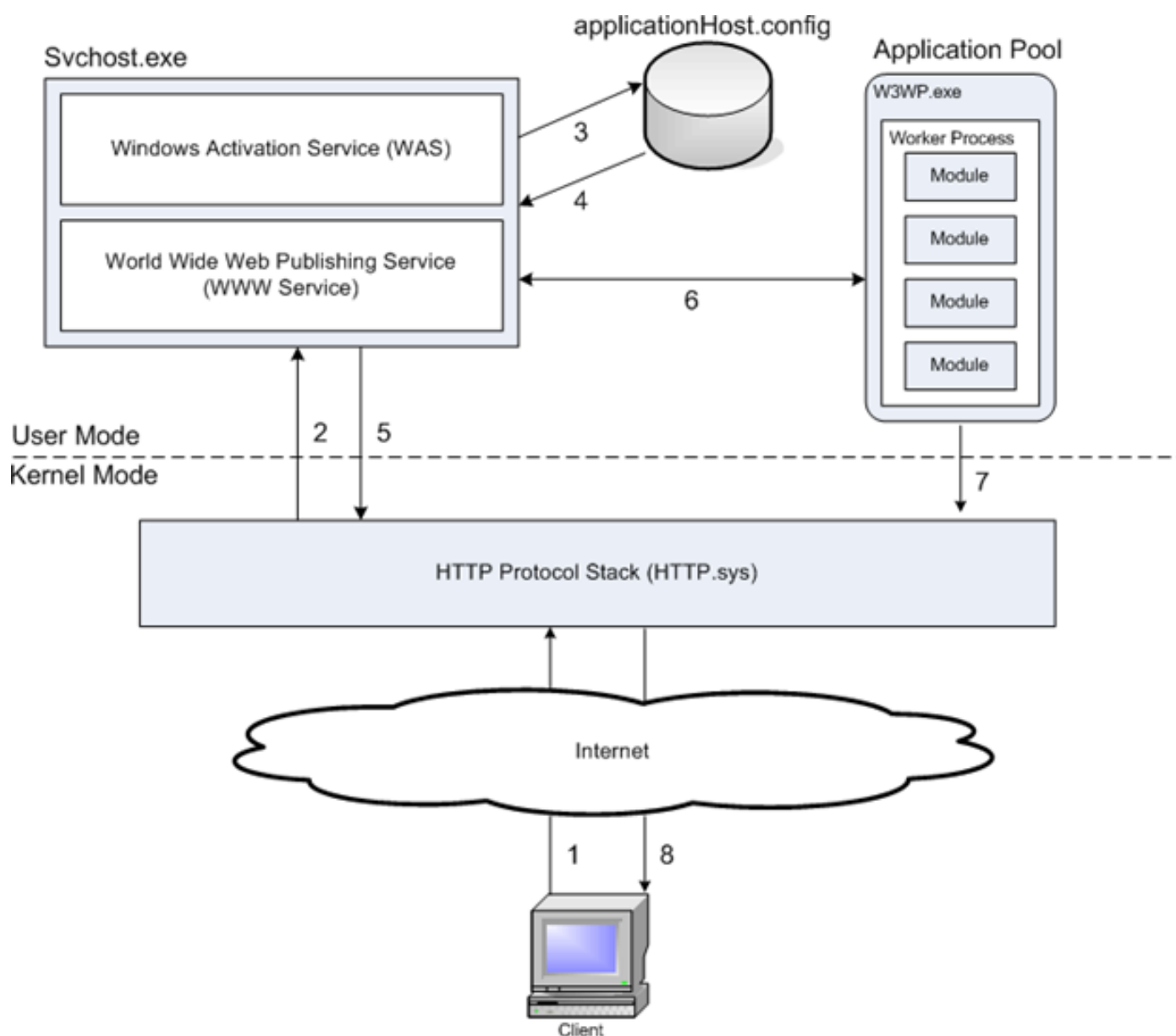
WAS هم اطلاعات پیکربندی شده را از محل ذخیره داده‌ها که applicationHost.config هست، می‌خواند. WWW Service که یک آداپتور برای Http.sys هست، اطلاعات را از WAS دریافت می‌کند. این اطلاعات شامل پیکربندی application pool و سایت می‌باشد.

WWW Service اطلاعات را برای Http.sys می‌فرستد.

WAS یک پروسه کارگر را در application pool ایجاد می‌کند تا درخواست رسیده مورد پردازش قرار بگیرد.

پروسه‌های کارگر درخواست را پردازش کرده و خروجی یا response مورد نظر را تولید می‌کنند.

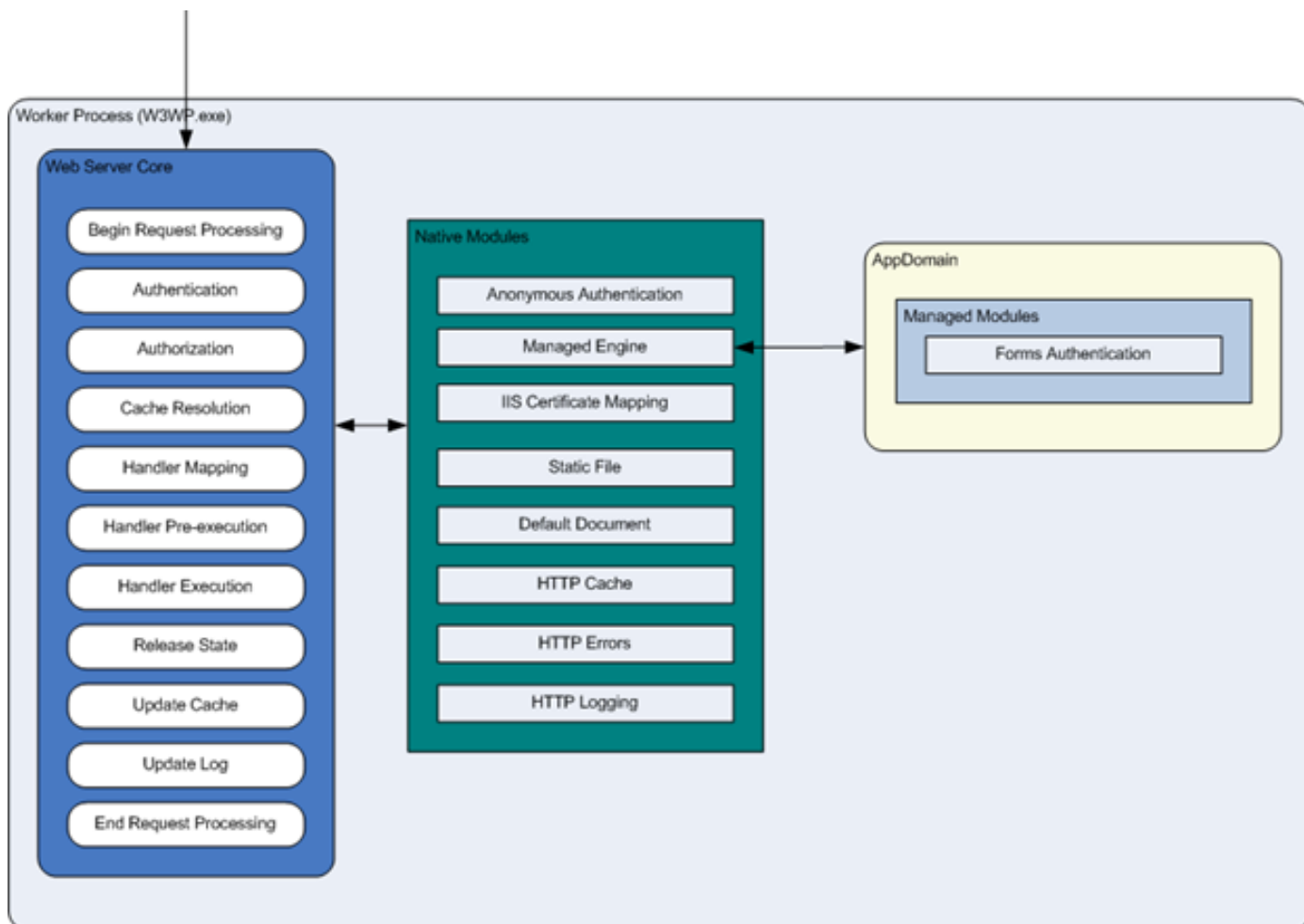
Http.sys نتیجه را دریافت و برای کلاینت می‌فرستد.



حال بیایید ببینیم موقعی که درخواست وارد پروسه‌ی کارگر میشود چه اتفاقی می‌افتد؟

در پروسه‌های کارگر، یک درخواست از مراحل لیست شده‌ای به ترتیب عبور می‌کند. در هسته وب سرور، رویدادهایی را فراخوانی می‌کند که در هر رویداد چندین ماژول native برای کارهایی چون authentication یا events logs دارد و در صورتیکه درخواستی نیاز به یک ماژول مدیریت شده CLR داشته باشد، از ماژول native managedEngine کمک گرفته و یک app domain را ایجاد می‌کند تا ماژول‌های مدیریت شده، عملیات لازم خودشان را انجام دهند. مثل authentication form و ...

موقعی هم که درخواست، از تمامی این رویدادها عبور کند، response برای http.sys ارسال می‌شود تا به کلاینت بازگشت داده شود. شکل زیر نحوه ورود یک درخواست به پروسه کارگر را نشان می‌دهد.



از نسخه 7 به بعد، IIS از یک معماری ماژولار استفاده می‌کند و این ویژگی، سه فایده دارد:

Componentization یا کامپوننت سازی
Extensibility یا توسعه پذیری یا قابل گسترش

ASP.NET Integration

Componentization

همه خصوصیات و ویژگی‌های این وب سرور، توسط کامپوننت‌ها مدیریت می‌شوند که باعث می‌شود شما به راحتی بتوانید کامپوننتی را اضافه، حذف یا جایگزین کنید و این باعث می‌شود که چندین امتیاز از IIS قبلی جلوتر باشد: باعث کاهش [attack surface](#) می‌شود که در نتیجه امنیت سیستم را بالا می‌برد. با ویژگی حذف کامپوننت‌ها شما می‌توانید ویژگی‌های غیرقابل استفاده IIS را حذف کنید تا ورودی‌های سیستم کاهش یابد. پس با کاهش ویژگی‌هایی که از آن هرگز استفاده نخواهید کرد، مدخل ورود هکر را از بین برده تا امنیت سرور بالاتر برود. افزایش کارایی و کاهش مصرف حافظه. با حذف ویژگی‌هایی که هرگز استفاده نمی‌کنید، در مصرف حافظه و بهینه استفاده شدن منابع سرور صرفه جویی کنید. با وجود ویژگی افزودن و جایگزینی کامپوننت‌ها، ناخودآگاه ذهن ما به سمت کاستوم سازی یا خصوصی سازی کشیده می‌شود. با این کار شما به راحتی یک custom server ایجاد می‌کنید که این سرور بر اساس علایق شما کارش را انجام می‌دهد و به راحتی امکاناتی چون افزودن third party را به توسعه دهنده می‌دهد.

Extensibility

با توجه به موارد بالا، خصوصی سازی باعث گسترش امکانات IIS می شود که می تواند به دلایل زیر اتفاق بیفتد:

قدرت بخشی به برنامه های وب. امکانات و قدرتی که می تواند در این حالت به برنامه های در حال اجرا داد به مراتب بیشتر از استفاده از لایه های داخلی خود برنامه هست. برای اینکار شما می توانید کدهای خود را با ASP.Net نوشته یا از کدهای native چون C++ استفاده کنید.

تجربه ای از توسعه پذیری ساده تر و راحت تر

استفاده از قدرت و تمامی امکانات را به شما می دهد و می توانید تمام دستورات را برای همه منابع حتی فایل های ایستا، ASP، CGI و دیگر منابع اجرا کنید.

ASP.NET Integration

تمامی موارد گفته شده بالا در این گزینه خلاصه می شود : محیط ASP.Net Integration به شما امکان استفاده از تمامی امکانات و منابع را به طور کامل می دهد. [دانلود ماژول های مدیریت شده](#)

[دانلود ماژول های native](#)