**<https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec>**

**What Is Service-Oriented Architecture?**

Service-Oriented Architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. Its principles are independent of vendors and other technologies. In service oriented architecture, a number of services communicate with each other, in one of two ways: through passing data or through two or more services coordinating an activity. This is just one definition of Service-Oriented Architecture. An article on Wikipedia goes into much more detail.

**معماری سرویس گرا چیست؟**  
معماری سرویس گرا (SOA) سبکی از طراحی نرم افزار است که در آن سرویس ها از طریق یک پروتکل ارتباطی از طریق شبکه به سایر مؤلفه ها ارائه می شوند. اصول آن مستقل از فروشندگان و سایر فناوری ها است. در معماری سرویس گرا ، تعدادی از خدمات به یکی از دو طریق با یکدیگر ارتباط برقرار می کنند: از طریق انتقال داده یا از طریق دو یا چند سرویس که یک فعالیت را هماهنگ می کنند. این فقط یک تعریف از معماری سرویس گرا است. مقاله ای در ویکی پدیا با جزئیات بیشتری همراه است.

**Characteristics Of Service-Oriented Architecture**

While the defining concepts of Service-Oriented Architecture vary from company to company, there are six key tenets that overarch the broad concept of Service-Oriented Architecture. These core values include:

Business value

Strategic goals

Intrinsic inter-operability

Shared services

Flexibility

Evolutionary refinement

Each of these core values can be seen on a continuum from older format distributed computing to Service-Oriented Architecture to cloud computing (something that is often seen as an offshoot of Service-Oriented Architecture).

**ویژگی های معماری سرویس گرا**  
در حالی که مفاهیم تعریف معماری سرویس گرا از یک شرکت به شرکت دیگر متفاوت است ، شش اصطلاح اصلی وجود دارد که مفهوم گسترده معماری سرویس گرا را برتری می دهد. این مقادیر اصلی شامل:  
  
ارزش تجارت  
اهداف استراتژیک  
قابلیت همکاری ذاتی  
خدمات به اشتراک گذاشته شده  
انعطاف پذیری  
پالایش تکاملی  
  
هر یک از این مقادیر اصلی را می توان در قالب مداوم از قالب قدیمی که توزیع محاسبات به معماری سرویس گرا تا محاسبات ابری توزیع می شود (چیزی که اغلب به عنوان شاخه ای از معماری سرویس گرا دیده می شود) مشاهده کرد.

[**https://en.wikipedia.org/wiki/Service-oriented\_architecture**](https://en.wikipedia.org/wiki/Service-oriented_architecture)

**معماری سرویس‌گرا** )به [انگلیسی](https://fa.wikipedia.org/wiki/%D8%B2%D8%A8%D8%A7%D9%86_%D8%A7%D9%86%DA%AF%D9%84%DB%8C%D8%B3%DB%8C): Service-oriented Architecture (به اختصار *SOA*))، یکی از رهیافت‌های طراحی نرم‌افزار با تمرکز بر طراحی [سامانه‌های توزیع‌شده](https://fa.wikipedia.org/wiki/%D8%B3%D8%A7%D9%85%D8%A7%D9%86%D9%87_%D8%AA%D9%88%D8%B2%DB%8C%D8%B9%E2%80%8C%D8%B4%D8%AF%D9%87) است. در این معماری کارکردهای نرم‌افزاری در قالب [سرویس](https://fa.wikipedia.org/wiki/%D8%B3%D8%B1%D9%88%DB%8C%D8%B3) توسط مؤلفه‌های برنامه‌های کاربردی به دیگر مؤلفه‌ها در بستر ارتباطات تحت شبکه ارائه می‌شود.[[۱]](https://fa.wikipedia.org/wiki/%D9%85%D8%B9%D9%85%D8%A7%D8%B1%DB%8C_%D8%B3%D8%B1%D9%88%DB%8C%D8%B3%E2%80%8C%DA%AF%D8%B1%D8%A7#cite_note-:0-1) [[۲]](https://fa.wikipedia.org/wiki/%D9%85%D8%B9%D9%85%D8%A7%D8%B1%DB%8C_%D8%B3%D8%B1%D9%88%DB%8C%D8%B3%E2%80%8C%DA%AF%D8%B1%D8%A7#cite_note-2)

امروزه از این معماری در دستگاه‌های دولتی و شرکت‌های خصوصی برای توسعه برنامه‌های کاربردی و سیستم‌ها، یکپارچگی سیستم‌های اطلاعاتی سازمانی یا تعاملات اطلاعاتی بین سازمانی و استاندارد وب سرویس (Web Service) و پروتکل‌های آن استفاده می‌شود

### پروتکل‌های معماری سرویسگرا

برخی از قرار داده‌های معماری سرویسگرا عبارتند از:

* [(**SOAP** Simple Object Access Protocol](https://fa.wikipedia.org/wiki/%D9%BE%D8%B1%D9%88%D8%AA%DA%A9%D9%84_%D8%AF%D8%B3%D8%AA%D8%B1%D8%B3%DB%8C_%D8%A2%D8%B3%D8%A7%D9%86_%D8%A8%D9%87_%D8%A7%D8%B4%DB%8C%D8%A7%D8%A1)): ساختاری برای تبادل پیام‌ها در قالب XML است که بین سرویس‌های وب تبادل می‌شود.
* [(**WSDL** Web service Description Language](https://en.wikipedia.org/wiki/Web_Services_Description_Language)): زبانی مبتنی بر XML که جهت توصیف ویژگی‌های عملیاتی سرویس‌های وب استفاده می‌شود و دارای دو بخش تعریف واسط و پیاده‌سازی است.
* [(**UDDI** Universal Description ,Discovery and Integration](https://en.wikipedia.org/wiki/Web_Services_Discovery)): واسطی است برای انتشار و شناسایی سرویس‌های وب و شامل یک مخزن می‌شود که ارائه دهندگان به انتشار و تبلیغ سوریس خود می‌پردازند تا دیگران بتوانند آن را شناسایی کنند.

### خواص معماری سرویسگرا

مهمترین خواص معماری سرویسگرا عبارتند از

* استفاده از استانداردهای مستقل از فناوری و مورد توافق برای ارائه مؤلفه‌های نرم‌افزاری تحت قالب سرویس
* معرفی‌کننده یک روش مشخص و مورد توافق برای تعریف و ارتباط بین مؤلفه‌های نرم‌افزاری
* مؤلفه‌های نرم‌افزاری منفرد می‌توانند در ساخت دیگر نرم‌افزارها استفاده شوند
* تقویت‌کننده رهیافت سرهم بندی اجزاء از قبل تعریف شده برای ساخت نرم‌افزارها به جای توسعه و پیاده‌سازی آنها
* می‌تواند به نرم افزاهای خارج سازمانی نیز مانند انواع داخلی آن متصل شوند.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

There are three roles in each of the Service-Oriented Architecture building blocks: service provider; service broker, service registry, service repository; and service requester/consumer.

در هر یک از بلوک های ساختمان معماری سرویس گرا سه نقش وجود دارد: ارائه دهنده خدمات؛ کارگزار خدمات ، ثبت خدمات ، مخزن خدمات؛ و درخواست کننده خدمات / مصرف کننده

The service provider works in conjunction with the service registry, debating the whys and hows of the services being offered, such as security, availability, what to charge, and more. This role also determines the service category and if there need to be any trading agreements.

ارائه دهنده خدمات در کنار رجیستری سرویس کار می کند ، درمورد نحوه و نحوه ارائه خدمات از قبیل امنیت ، در دسترس بودن ، چه چیزی شارژ و موارد دیگر بحث می کند. این نقش همچنین دسته بندی خدمات را تعیین می کند و در صورت نیاز به توافق نامه تجارت وجود دارد.

The service broker makes information regarding the service available to those requesting it. The scope of the broker is determined by whoever implements it.

کارگزار خدمات اطلاعات مربوط به این سرویس را در اختیار کسانی که درخواست آن می شوند ، می کند. دامنه کارگزار توسط هرکسی که آن را پیاده کند تعیین می شود.

The service requester locates entries in the broker registry and then binds them to the service provider. They may or may not be able to access multiple services; that depends on the capability of the service requester.

درخواست کننده خدمات ، ورودی هایی را در فهرست کارگزاران پیدا کرده و سپس آنها را به ارائه دهنده خدمات متصل می کند. آنها ممکن است یا نتوانند به چندین سرویس دسترسی پیدا کنند. این به توانایی درخواست کننده خدمات بستگی دارد.

# Implementing Service-Oriented Architecture

When it comes to implementing service-oriented architecture (SOA), there is a wide range of technologies that can be used, depending on what your end goal is and what you’re trying to accomplish.

وقتی نوبت به اجرای معماری سرویس گرا (SOA) می رسد ، بسته به اینکه هدف نهایی شما چیست و سعی در دستیابی به آن دارید ، طیف گسترده ای از فناوری ها قابل استفاده هستند.

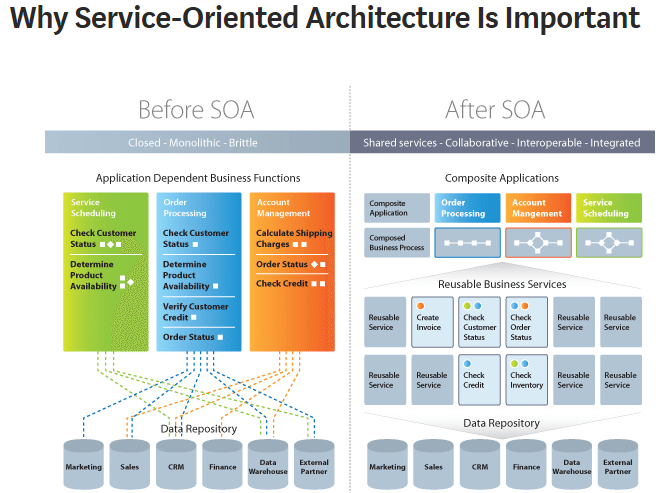
Typically, Service-Oriented Architecture is implemented with web services, which makes the “functional building blocks accessible over standard internet protocols.”

به طور معمول ، معماری سرویس گرا با خدمات وب پیاده سازی می شود ، که باعث می شود "بلوک های ساختاری کاربردی بر روی پروتکل های استاندارد اینترنت قابل دسترسی باشند."

An example of a web service standard is [SOAP](https://en.wikipedia.org/wiki/SOAP), which stands for Simple Object Access Protocol. In a nutshell, SOAP “is a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks. Although SOAP wasn’t well-received at first, since 2003 it has gained more popularity and is becoming more widely used and accepted. Other options for implementing Service-Oriented Architecture include Jini, COBRA, or REST.

نمونه ای از استاندارد سرویس وب SOAP است که مخفف پروتکل دسترسی ساده به شی است. به طور خلاصه ، SOAP "مشخصات پروتکل پیام رسانی برای تبادل اطلاعات ساختاری در اجرای سرویس های وب در شبکه های رایانه ای است. اگرچه SOAP در ابتدا مورد استقبال خوبی قرار نگرفت ، از سال 2003 محبوبیت بیشتری پیدا کرد و بیشتر مورد استفاده و پذیرش قرار می گیرد. گزینه های دیگر برای اجرای معماری سرویس گرا شامل Jini ، COBRA یا REST است.

It’s important to note that architectures can “operate independently of specific technologies,” which means they can be implemented in a variety of ways, including messaging, such as ActiveMQ; Apache Thrift; and SORCER.

توجه به این نکته مهم است که معماری ها می توانند "مستقل از فن آوری های خاص" عمل کنند ، به این معنی که آنها می توانند به روش های مختلفی از جمله پیام رسانی مانند ActiveMQ اجرا شوند؛ Apache Thrift؛ و SORCER

.

There are many [benefits to service-oriented architecture](https://www.bmc.com/blogs/service-oriented-architecture-overview/), especially in a web service based business. We’ll outline a few of those benefits here, in brief:

برای معماری سرویس گرا مزایای بسیاری دارد ، خصوصاً در یک تجارت مبتنی بر سرویس. ما به اختصار چند مورد از این مزایا را شرح خواهیم داد:

Use Service-Oriented Architecture to create reusable code: Not only does this cut down on time spent on the development process, but there’s no reason to reinvent the coding wheel every time you need to create a new service or process. Service-Oriented Architecture also allows for using multiple coding languages because everything runs through a central interface.

برای ایجاد کد قابل استفاده مجدد از معماری سرویس گرا استفاده کنید: این کار نه تنها در زمان صرف شده در روند توسعه کاهش می یابد بلکه هیچ دلیلی برای ایجاد مجدد چرخ کدگذاری در هر بار نیاز به ایجاد یک سرویس یا فرآیند جدید وجود ندارد. معماری سرویس گرا همچنین امکان استفاده از چندین زبان رمزگذاری را فراهم می آورد زیرا همه چیز از طریق رابط مرکزی اجرا می شود.

Use Service-Oriented Architecture to promote interaction: With Service-Oriented Architecture, a standard form of communication is put in place, allowing the various systems and platforms to function independent of each other. With this interaction, Service-Oriented Architecture is also able to work around firewalls, allowing “companies to share services that are vital to operations.”

از معماری سرویس گرا استفاده کنید تا تعامل را ارتقا دهید: با معماری سرویس گرا ، یک شکل استاندارد از ارتباط برقرار می شود که به سیستم ها و سیستم عامل های مختلف اجازه می دهد تا مستقل از یکدیگر کار کنند. با این تعامل ، معماری سرویس گرا همچنین قادر به کار در اطراف دیوارهای آتش است و به شرکت ها امکان می دهد سرویس هایی را که برای عملیات ضروری هستند به اشتراک بگذارند.

Use Service-Oriented Architecture for scalability: It’s important to be able to scale a business to meet the needs of the client, however certain dependencies can get in the way of that scalability. Using Service-Oriented Architecture cuts back on the client-service interaction, which allows for greater scalability.

برای مقیاس پذیری از معماری سرویس گرا استفاده کنید: مهم است که بتوانید یک تجارت را برآورده سازید تا نیاز مشتری را برآورده سازید ، اما وابستگی های خاصی می تواند به این مقیاس پذیری برسد. استفاده از معماری سرویس گرا ، تعامل مشتری و مشتری را کاهش می دهد ، که امکان مقیاس پذیری بیشتری را فراهم می آورد.

Use Service-Oriented Architecture to reduce costs: With Service-Oriented Architecture, it’s possible to reduce costs while still “maintaining a desired level of output.” Using Service-Oriented Architecture allows businesses to limit the amount of analysis required when developing custom solutions.

از معماری سرویس گرا استفاده کنید تا هزینه ها را کاهش دهید: با معماری سرویس گرا ، می توانید هزینه ها را کاهش دهید در حالی که هنوز "سطح مطلوب خروجی را حفظ می کنید". استفاده از معماری سرویس گرا به مشاغل اجازه می دهد تا میزان تجزیه و تحلیل مورد نیاز را هنگام تهیه راه حل های سفارشی محدود کنند.