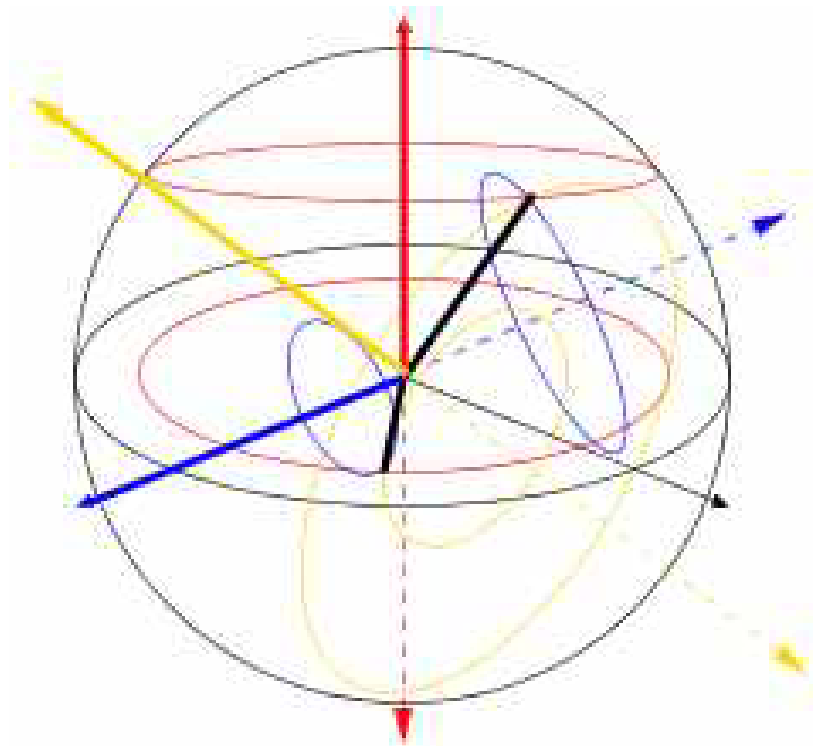


# Application Oriented Networks: An SOA Perspective



# Introduction

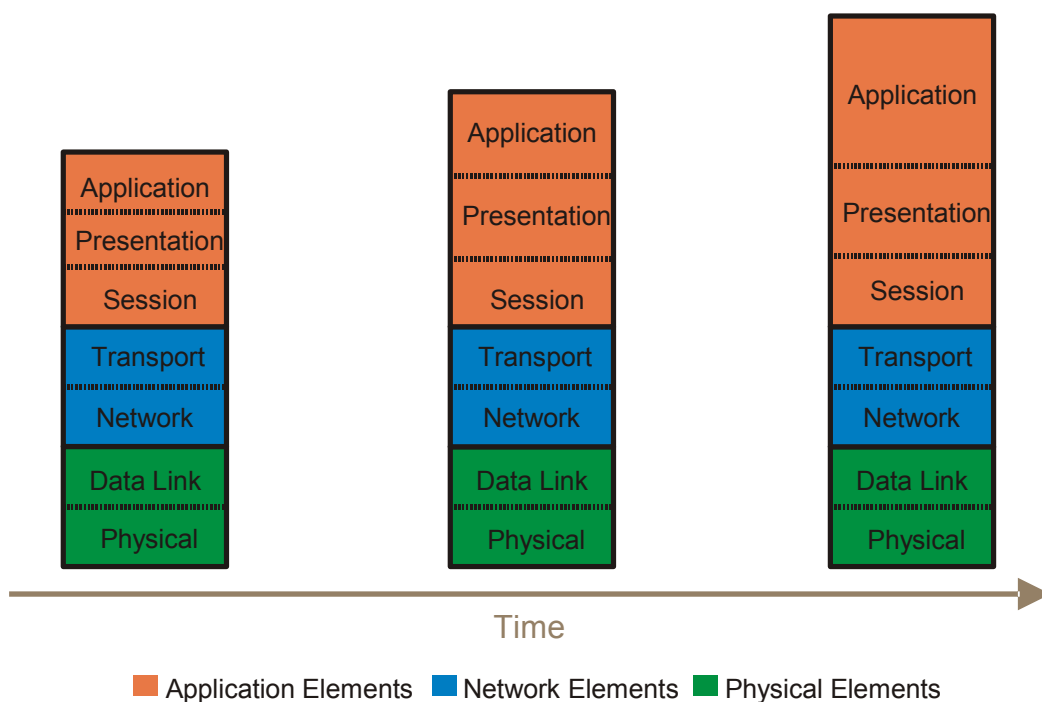
Service Oriented Architecture is the hot topic of discussion in IT circles today. So much so, in fact, that SOA is being seen by many as the future of enterprise interaction. Even as SOA continues to make waves, newer and more radical methods of implementing it are emerging every day. The field is so vast that there is a lot of room for new players.

The concept of 'Application Oriented Networks' is one of many technologies that is being considered for use in an SOA environment. Developed almost independently as a concept, AON is quickly beginning to look like one of the strongest contenders for use with the SOA paradigm. This whitepaper explores the facts and attempts to analyze the true impact of AON on SOA.

# AON

The concept of AON is the realization of the “smart network” dream. Simply put, Application Oriented Networks go beyond the standard TCP/IP networks by making the individual network elements more intelligent.

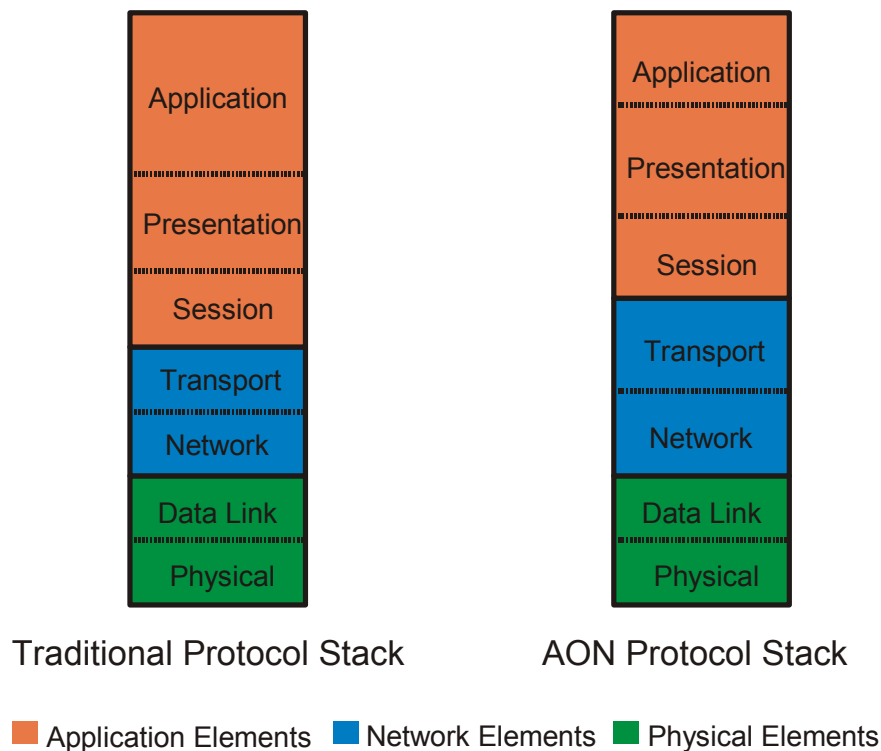
The evolution of AONs came as a solution to the problem of increasing complexity in the Application Layer of the traditional protocol stack, while the Network and Transport Layers continued to be relatively simple.



An extremely complex Application Layer has a vast set of attendant problems, some of the major ones being:

- Lower speed owing to large application messages that span multiple MTUs (Maximum Transmission Units).
- Higher cost of development and maintenance.
- A constant need for more and more middleware to integrate two vastly different applications.

By moving some of this complexity out of the Application layer and into the Network and Transport Layers, AONs seek to balance the protocol stack to a certain extent.



The new developments in this space allow AONs to provide a wide variety of functions including in-line security, transformation, routing and business event awareness.

With a growing need for platform and vendor independence, XML has become the de-facto standard for enterprise interactions. XML's platform independent nature and adaptability to diverse applications have been the basis of its growing popularity. It is no surprise then that the most obvious method of adding Application layer functionality to the network layer is to make the individual network elements such as gateways and routers "XML aware". This could be by way of add-on hardware or by building the elements to inherently contain more functionality.

By adding on this intelligence at the network level, AONs could help achieve some of the most sought after goals in the enterprise software framework, such as:

- \* Fine-grained yet low-cost application infrastructure
- \* Pervasive business event awareness
- \* Increased performance
- \* Comprehensive security

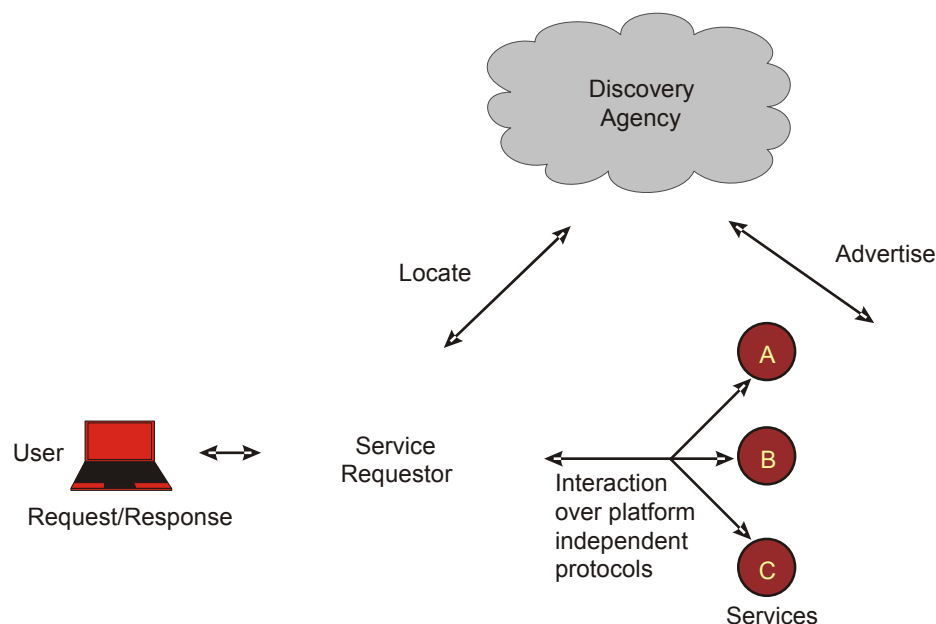
# SOA

Before we analyze the application of AONs to Service Oriented Architectures, it is necessary to look at the basics of SOA and to understand the underlying motivation behind the evolution of SOAs.

The Service Oriented Architecture paradigm evolved from the principles of Object Oriented Programming and component based software, when developers began to realize that merely re-using components was not enough to provide maximum efficiency in a distributed environment. The need for application integration and interoperability formed the basis for SOA.

SOA has caught the fancy of enterprise architects worldwide, since it enables businesses to respond to change quickly and effectively. This is achieved by having a large number of finely grained services, defined at a level of granularity that can be orchestrated in different patterns to achieve diverse tasks.

The key players in an SOA are the Service Requestor, the Service Broker and the Service Provider, as depicted in the diagram below. Service Providers publish their services to the world by way of the Service Broker. The Service Requestor can then send a request based on need, to the Service Broker, who will in turn return the service that most closely matches the requirement of the requestor, from its directory.



SOA is being heralded as the latest step in the evolution of middleware. SOA goes beyond the basic middleware in its fundamental principles, which are:

- \* Loose coupling
- \* High interoperability
- \* Re-usability
- \* Fine-grained service definitions

## Areas of Intersection

From the above sections, it is a fairly intuitive deduction that AONs can contribute significantly to SOAs.

AONs with their ability to understand complex, high-level application message formats, allow far more flexibility in application design than any traditional network. In order for SOA to enable agile business processes, the individual services need to be extremely fine grained. This is to ensure that they can be used in a wide variety of business processes without any change. The function of the service needs to be atomic, to minimize redundancy in the service orchestration pattern.

Fine-grained services also ease the process of end to end testing and tend to be more re-usable. However, when the number of services with fine-grained definitions increase, the number of interactions between different services to accomplish a single business process also increases. In this increased interaction scenario it is important to keep in mind:

- \* Security, since increased interaction also means increased vulnerability.
- \* Performance, since each interaction will have a finite latency.
- \* Interoperability, since finer grained service definitions also leave more room for perpetrating tightly coupled interfaces.

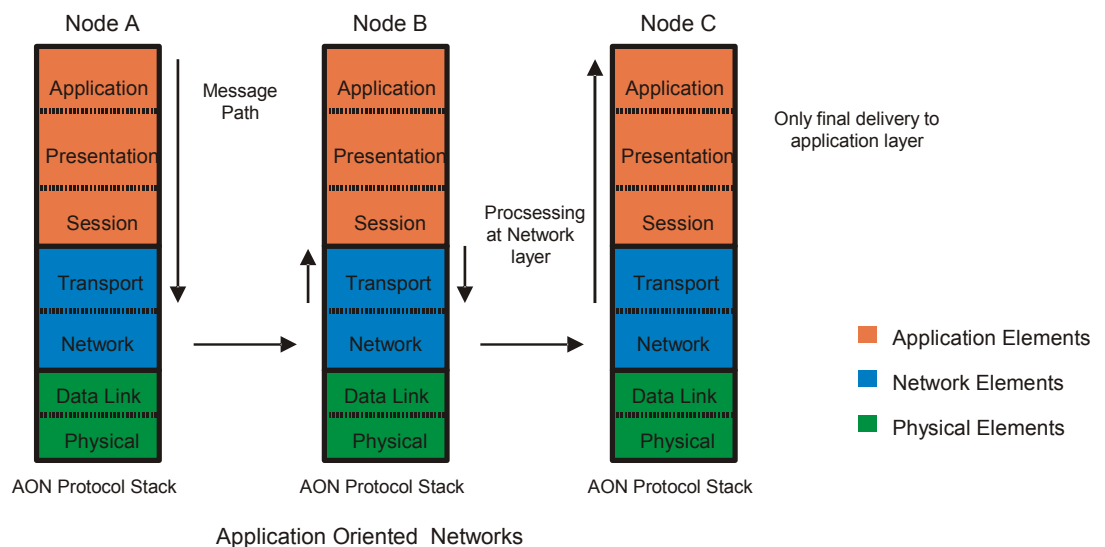
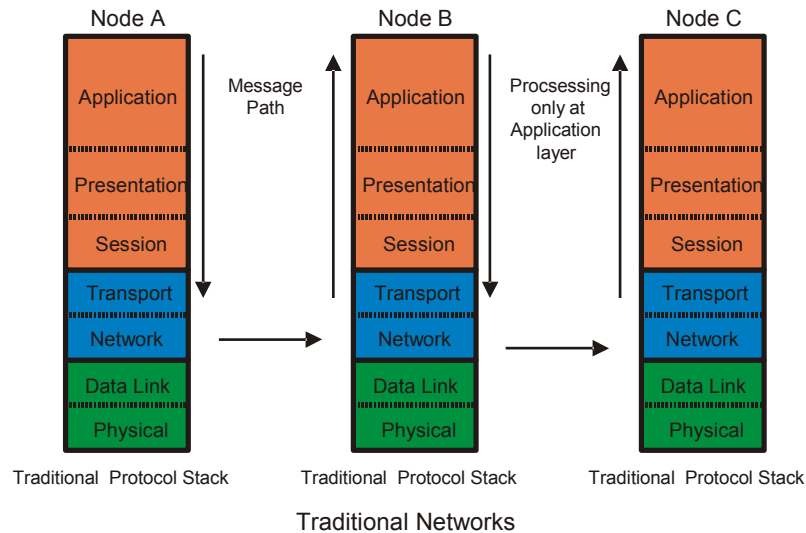
AONs can help in all three of these areas:

Security in an AON can be more comprehensive and pervasive than in a traditional network, since it can be implemented at the network device level. With the new breed of AON specific “smart network” devices, security and authentication become much simpler to implement and easier to manage.

Performance in an AON increases radically, primarily on account of AONs being application aware. As a result of this, priority scheduling and routing, dynamic routing and congestion control, etc. become much easier to implement. Performance bottlenecks can be broken down from a high level, application specific point of view, rather than at the packet level.

Since the AON is application aware, the number of traversals up and down the protocol stack is significantly reduced. For example in a traditional environment, messages from the Application layer can only be interpreted by another Application layer. Traversing the middleware, therefore, takes longer, since the message must travel right up the protocol stack to the application layer and then be repackaged for forwarding.

In an AON, it would be possible to do the repackaging at the network level. The only time the data would need to reach the application layer would be in cases where some processing would be required.



Interoperability and loose coupling are highly important, particularly with the danger of perpetrating tightly coupled interfaces as a response to the need for fine-grained service definitions. This is a concern, particularly in cases where component based software is in use, which may have rigidly defined inter-component communication mechanisms. When the functionality of these components is required to be exposed as a service, the coupling between the resulting services is often tight.

With AONs, since the network is 'application aware' and typically, platform independent, the probability of tightly coupled interfaces being created is reduced, since the interaction will be managed at the network level, which has the requisite knowledge required for such management.

In addition to this, AONs also provide business event visibility to the network layer; making orchestration smoother and providing scope for development in yet unexplored areas. For example, if a business event occurs, which requires a directional change in the routing of data, the turnaround time required to alter the flow of messages can be reduced in case of an AON, since the business event can be analyzed and the appropriate routing changes made at the Network level instead of waiting for an Application level decision.

Finally, AONs create the possibility of exposing a lot of network functionality as services. For example, the congestion control functionality could be exposed as a service, with an instance on each network device, to ensure that the flow of information remains smooth and uniform. Similarly, integrity checking and validation on messages could be done at the network level instead of having the message traverse the stack at each node merely to be verified.

## The Road Ahead

The possibilities that AONs open up for the SOA paradigm are endless and almost completely unexplored. The ability to transfer a large chunk of functionality from the Application layer to the Network layer could mean substantial changes in the way businesses use IT. The obvious advantages are application and service architectures that are cheaper to build and faster to deploy, since much of the redundant functionality can be isolated and made an implicit part of the network itself.

In addition to this, since smarter networks mean better performance, despite more interactions between services, the constraints on finely grained service definitions become less restrictive, leading to more efficiently designed SOAs.

In summation, AONs could well be one of the key concepts to help SOA realize its full potential.

Torry Harris Business Solutions (THBS) is a US based IT service provider with development facilities in India and China. The company, started in 1998, has for several years delivered a large variety of middleware services to enterprise clients around the world. Now, with a large pool of highly skilled technologists and rapidly growing, the company remains focused on the middleware and integration space, implementing large projects across the US, Europe, the Middle East and the Far East. The company is committed to Service-oriented Architecture(SOA), which it sees as the logical movement to follow the phenomenon of distributed computing in the late nineties, where THBS was clearly the market leader in implementing the offshore/onsite delivery model.

For further information, write to us at [soa@thbs.com](mailto:soa@thbs.com).