



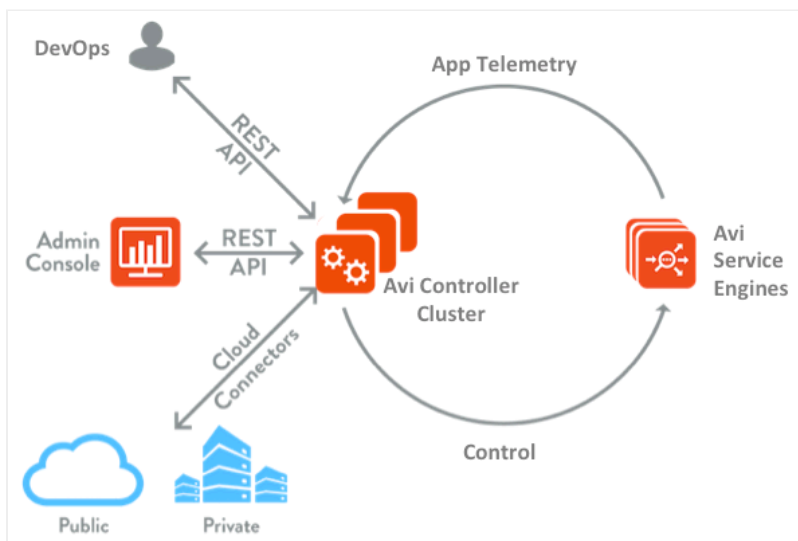
**Avi Vantage Platform
| Architectural
Overview | Knowledge
Base**

Avi Technical Reference (v17.2)

Avi Vantage Platform | Architectural Overview | Knowledge Base

[view online](#)

The Avi Vantage platform is built on software-defined principles, enabling a next generation architecture to deliver the flexibility and simplicity expected by IT and lines of business. The Avi Vantage architecture separates the data and control planes to deliver application services beyond load balancing, such as application analytics, predictive autoscaling, micro-segmentation, and self-service for app owners in both on-premises or cloud environments. The platform provides a centrally managed, dynamic pool of load balancing resources on commodity x86 servers, VMs or containers, to deliver granular services close to individual applications. This allows network services to scale near infinitely without the added complexity of managing hundreds of disparate appliances.



The Avi Controller cluster uses big data analytics to analyze the data and present actionable insights to administrators on intuitive dashboards on the Avi Admin Console.

Avi Vantage provides out-of-the-box integrations for on-premises or cloud deployments. These integrations with private cloud frameworks, SDN controllers, container orchestration platforms, virtualized environments and public clouds enable turnkey application services and automation.

Avi Vantage Components

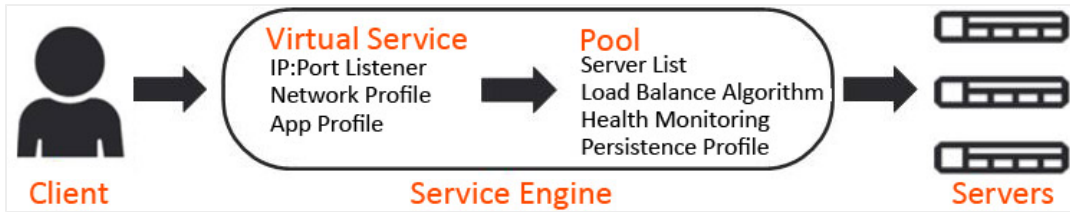
The Avi Vantage Platform has three core components ? Avi Service Engines, Avi Controller cluster, and Avi Admin Console:

Service Engine

Avi Service Engines (SEs) handle all data plane operations within Avi Vantage by receiving and executing instructions from the Controller. The SEs perform load balancing and all client- and server-facing network interactions. It collects real-time application telemetry from application traffic flows. High availability is supported.

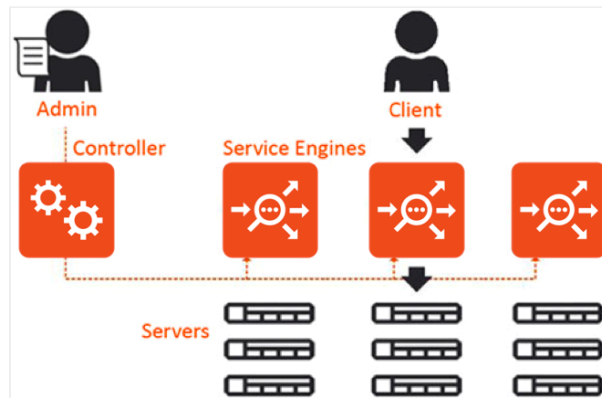
In a typical load balancing scenario, a client will communicate with a virtual service, which is an IP address and port hosted in Avi Vantage by an SE. The virtual service internally passes the connection through a number of profiles. For HTTP traffic, the SE may terminate and proxy the client TCP connection, terminate SSL, and proxy the HTTP request. Once the request has been validated, it will be forwarded internally to a pool, which will choose an available server. A new TCP connection then

originates from the SE, using an IP address of the SE on the internal network as the request’s source IP address. Return traffic takes the same path back. The client communicates exclusively with the virtual service IP address, not the real server IP.



Controller

The Avi Controller is a single point of management and control that is the ‘brain’ of the entire Avi Vantage system, and typically deployed as a redundant three-node cluster. The entire Avi Vantage system is managed through a centralized point (and IP address) regardless of the number of new applications being load balanced and the number of SEs required to handle the load. Via its REST API, it provides visibility into all applications configured. Controllers can automatically create and configure new SEs as new applications are configured via virtual services (in write access mode deployments).



Controllers continually exchange information securely with the SEs and with one other. The server health, client connection statistics, and client-request logs collected by the SEs are regularly offloaded to the Controllers, which share the processing of the logs and analytics information. The Controllers also send commands down to the SEs, such as configuration changes. Controllers and SEs communicate over their management IP addresses. (Click [here](#) for a list of the protocol ports Avi Vantage uses for management.)

Console

The Avi Console is a modern web-based user interface that provides role-based access to control, manage and monitor applications. Its capabilities are likewise available via the Avi CLI. All services provided by the platform are available as REST API calls to enable IT automation, developer self-service, and a variety of third party integrations.

Data Plane Scaling

Virtual services may be scaled across one or more SEs using either [native](#) or [BGP-based](#) load balancing techniques.

When a virtual service is scaled across multiple SEs, each of those SEs share the load. This sharing may not be equal, because the actual workload distribution depends on the available CPU and other resources that may be required of the SE. SEs typically process traffic for more than one virtual service at a time.



When [native SE scaling](#) is used, one SE will be the primary for a given virtual service and will advertise that virtual service's IP address from the SE's own MAC address. The primary SE may either process and load balance a client connection itself, or it may forward the connection via layer 2 to the MAC address of one of the secondary SEs having available capacity.

Each SE will load balance and forward traffic to servers using its own IP address within the server network as the source IP address of the client connection. This ensures that even though multiple SEs may be sending traffic to the same pool of servers, return traffic takes the same path from the servers back through the same SE. When deployed in a VMware environment and the SEs are scaled out, the secondary SEs will respond directly back to clients without passing the return traffic back through the primary SE. In OpenStack with native SE scaling, the secondary SEs will return client responses back through the primary SE.