

WILS: Controllers and Gateways



Lori MacVittie, 2010-29-11

Why these two are very different but complementary technologies

Have you ever wondered why one network product is called a “controller” while another seemingly similar in function solution is called a “gateway”? There’s actually a very good reason for the naming and despite *appearing* to act similarly they do fill different roles in an architecture and are often called upon to work together.

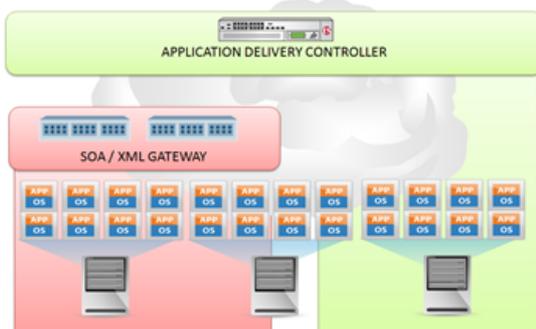
GATEWAYS

If you loosely defined a gateway as a “converter” or “translator” you’d be very close to nailing down a simple definition. Gateways act as mediators between disparate networks. In networking terms this is most commonly associated with routers (your “default gateway”) as it is designed as a transition point between two similar resources (networks) that generally utilize different protocols or in some way require a translation of messages/packets in order to traverse the network on the “other side” of the gateway. SOA (or XML) gateways, for example, provide a transition point between XML/Web Service endpoints and clients that can translate application messages from one format to another, either between different XML subsets or even between transport protocols: HTTP → JMS, for example.

Cloud gateways, and in particular cloud storage gateways of late, are products that mediate between enterprise users and applications (in some cases) and cloud-based storage services. These devices provide translation from an internal protocol such as SAMBA or CIFS to the specific cloud-based service “language” (API) required to access data stored in cloud-based locations.

CONTROLLER

A “controller” has traditionally referred to devices that control (hence the name) the transfer of data between a computer and its peripheral devices, and vice versa. The use of controller in the data center/architectural context means much the same, with the caveat that the “peripherals” involved are the endpoints, i.e. the client and the server(s). This is why some folks fly into a rage when hearing an “application delivery controller” referred to as a “load balancer”. An [application delivery controller](#) of course provides [load balancing](#) services but it also [controls the transfer of data](#) between the client and the servers (in myriad ways) as a means to provide other services as well, e.g. application and protocol security, acceleration, and optimization.



Network-deployed controllers generally have more intelligence and flexibility built into them in terms of directing the flow of data. For example, a file virtualization controller governs the storage of files (data) across a variety of storage tiers based on rules specified by the business or operations staff. An application delivery controller can make a wealth of decisions regarding where to direct an HTTP request again based on rules specified by the business or

operations staff.

DIFFERENT but COMPLEMENTARY

Often times an architecture will be comprised of both gateways and controllers, because the controllers often lack the translation capabilities of a gateway and the gateway lacks the intelligence and flexibility of the controller. Consider SOA gateways and application delivery controllers. It is often the case that a “farm” of gateways is front-ended (load balanced) by an application delivery controller. This is true of many types of “gateways” because if you think about, the process of transforming data and protocols is intense and these devices were not necessarily designed with scale and volume in mind, but rather functionality or, in the case of XML gateways with the scale and performance of a single type of process: parsing XML. They are not necessarily designed with network stack security or intelligent routing *other* than a specific, niche application.

Similarly, the beginnings of a [dual architecture comprising storage virtualization solutions and cloud storage gateways](#) is beginning to emerge. This is because the storage virtualization solution (the controller) has the intelligence to make decisions regarding whether files/data should be stored locally or remotely, in a cloud-based storage environment.

WILS: WRITE IT LIKE SETH. SETH GODIN ALWAYS GETS HIS POINT ACROSS WITH BREVITY AND WIT. WILS IS AN ATTEMPT TO BE CONCISE ABOUT APPLICATION DELIVERY TOPICS AND JUST GET STRAIGHT TO THE POINT. NO DILLY DALLYING AROUND.

Related blogs & articles:



- [I CAN HAS DEFINISHUN of SoftADC and vADC?](#)
- [Intercloud: The Evolution of Global Application Delivery](#)
- [The Application Delivery Deus Ex Machina](#)
- [Dynamic Intelligent Application Delivery in a Distributed ...](#)
- [Optimize Prime: The Self-Optimizing Application Delivery Network](#)
- [Can the future of application delivery networks be found in neural ...](#)
- [ROI Justification\(s\) for Application Delivery Controllers](#)
- [WILS: Load Balancing and Ephemeral Port Exhaustion](#)
- [All WILS Entries on DevCentral](#)
- [WILS: The Concise Guide to *-Load Balancing](#)
- [WILS: What Does It Mean to Align IT with the Business](#)



F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com