

Is Your Application Infrastructure Architecture Based on the Postal Service Delivery Model?



Lori MacVittie, 2010-06-01

If it is, you might want to reconsider how you're handling security, acceleration, and delivery of your applications before users "go postal" because of poor application performance.

Sometimes wisdom comes from the most unexpected places. Take [Jason Rahm's](#) status update on [Facebook](#) over the holidays. He's got what is likely a common complaint



Jason Rahm would like to take a crack at solving a few of the horrible inefficiencies of the postal service. Problem 1: why does the destination office decide whether the correct amount of postage has been paid? The entrance point should be the cost enforcer, any step after that costs everyone more money.

Yesterday at 1:16pm via Facebook for BlackBerry · Comment · Like

regarding the delivery model of the [US postal service](#): the inefficiency of *where* postage due is determined. Everyone has certainly had the experience of sending out a letter (you know, those *paper* things) and having it returned a week or more later with a big stamp across it stating: **Returned – Postage Due**.

As Jason points out, the US postal service doesn't determine whether postage may be due or not until the package arrives at its destination. If the addressee isn't willing/able to pay that postage due, the package is of course returned via the delivery service, which incurs round-trip costs of transportation and handling at every point along the way.

If this sounds anything like your application infrastructure architecture, then you might want to reconsider how you're handling the delivery of applications and where you're applying policies that may affect the delivery process.

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