

Joint Infringement and the Future of Interactive Methods and System Patents

W. Keith Robinson

Emerging consumer technologies focus on increasing connectivity between different devices and their users. What happens then when a patentee presents a court with asserted method claims that require performance by multiple parties? *BMC Resources, Inc. v. Paymentech, L.P.* and *Muniauction, Inc. v. Thomson Corp.* articulate the standards for joint infringement. However, recent Federal Circuit decisions in *Akamai Technologies, Inc. v. Limelight Networks, Inc.* and *McKesson Technologies, Inc. v. Epic Systems Corp.* have applied the standard in less than satisfying form. On April 20, 2011, this led the U.S. Court of Appeals for the Federal Circuit to agree to address the question of whether and under what circumstances there can be joint liability for patent infringement. The Federal Circuit granted Akamai's request for a rehearing *en banc* of the decision in *Akamai Technologies, Inc.*, where the entire court will consider under what circumstances there can be joint liability when two or more parties collectively perform all of the steps of a method claim but no one party performs all of the claim's steps.

This article analyzes recent developments concerning joint infringement theory which has a significant impact on how patents directed to emerging technologies are written and enforced. Past decisions have focused on how the asserted claims are drafted and the relationships between the accused infringer and third parties. This article suggests that the current direction or control standard is too restrictive and has required practitioners to draft awkward claims that do not fully capture interactive methods and systems. Instead, this article advocates for a more workable contributory infringement standard. In addition, this article cautions against holding infringers of interactive methods and systems jointly and severally liable for infringement and presents other options for fairly determining liability.