# Abstract

This paper describes a novel approach to providing modular and extensible operating system functionality, and encapsulated environments, based on a synthesis of microkernel and virtual machine concepts. We have developed a *virtualizable architecture* that allows recursive virtual machines (virtual machines running on other virtual machines) to be efficiently implemented, in software, by a microkernel running on generic hardware. A complete virtual machine interface is provided at each level; efficiency derives from needing to implement only *new* functionality at each level.

This infrastructure allows common OS functionality, such as process management, demand paging, fault tolerance, and debugging support, to be provided by cleanly modularized, independent, stackable virtual machine monitors, implemented as ordinary user processes. It can also provide uncommon or unique OS features, including the above features specialized for particular applications' needs, or virtual machines transparently distributed cross-node, or security monitors that allow arbitrary untrusted binaries to be safely executed. Our prototype implementation of this model indicates that it is practical to modularize operating systems this way: some types of virtual machine layers impose almost no overhead at all, while others impose some overhead (typically 10–20%), but only on certain classes of applications.