Problem Addressed :

Traditional operating systems provide a set of abstractions for applications to use physical resources. These high-level abstractions have a significant impact on applications - they reduce their performance, hide information about the implementation of resource management, and limit the functionality of these applications.

Key idea :

The paper proposes Exokernel, an OS architecture that has a minimal kernel, in which resources are provided to applications through the library operating system. Applications are securely binded to physical resources to use them, providing protection and separating authorization and use of resources. Visible resource revocation protocol is used by exokernel, to deallocate the resources from applications. Abort protocol is provided to take back the resource and break the secure binding from the application if the application fails to respond quickly.

Strengths :

* The exokernel design provides more extensibility and flexibility to the applications as it minimizes the abstractions provided by the OS. In this way, an application can make use of physical resources in its own way, without thinking about the implementation of resources by the OS.
* Library operating system provides specialized high-level abstractions which can optionally be used by applications. These abstractions meet the requirements of applications. Also, multiplexing of resources can be done efficiently.

Weakness :

* By not providing the abstractions, it is the responsibility of an application to write those abstractions for itself. This makes the application bigger and complex to make.
* This is just a concept paper. No real OS was created to do the experiment. Just a prototype using Aegis and ExOS was created. The results were on par with other general OS and did not show significant better performance.

Additional Thoughts :

This paper tried to solve the same issue of extensibility in OS as did the SPIN OS, however, using a different and rather a straightforward approach. I believe that using the concept of minimal kernel and secure bindings can help solve this problem, but I have reservations regarding the experiment and results shown in the paper. The results do not show any better performance than other architecture. Also, many security protocols were not implemented for the experiment and hence the results cannot be considered.