KESO • A Type-Safe Middleware for Embedded Systems

Domains are the realms of memory protection in KESO similar to processes in modern operating systems. KESO’s protection is built on Java’s type-safety combined with the strict separation of the object heaps of each domain and replication of mutable global data in each domain. The isolation inhibits the spreading of an error and allows the safe integration of different tasks on a node.

KESO enables you to relocate domains from one node to a different node in the system by only changing the configuration. Inter-domain communication continues to work without changing the application due to the uniform portal mechanism.

KESO is firmly based on an event-driven OSEK OS that is widely spread in the automotive industry. The priority-based scheduler of the OSEK OS is used for the scheduling of the tasks in a KESO system. Other features of the OSEK OS such as the priority-ceiling-based synchronization mechanisms are also provided by the KESO API to the Java applications, whereby KESO allows to enforce domain-based access restrictions.

KESO provides special memory objects that allow r/w access to a specific region of memory. Only primitive datatypes can be read and written through this interface. Range checks are performed on each access to prevent memory corruptions. Memory objects can also be used to provide shared memory between domains which allows the efficient exchange of large amounts of data.

Department of Computer Sciences
Distributed Systems and Operating Systems
http://www4.cs.fau.de/Research

Christian Wawersich • Michael Stilkerich