



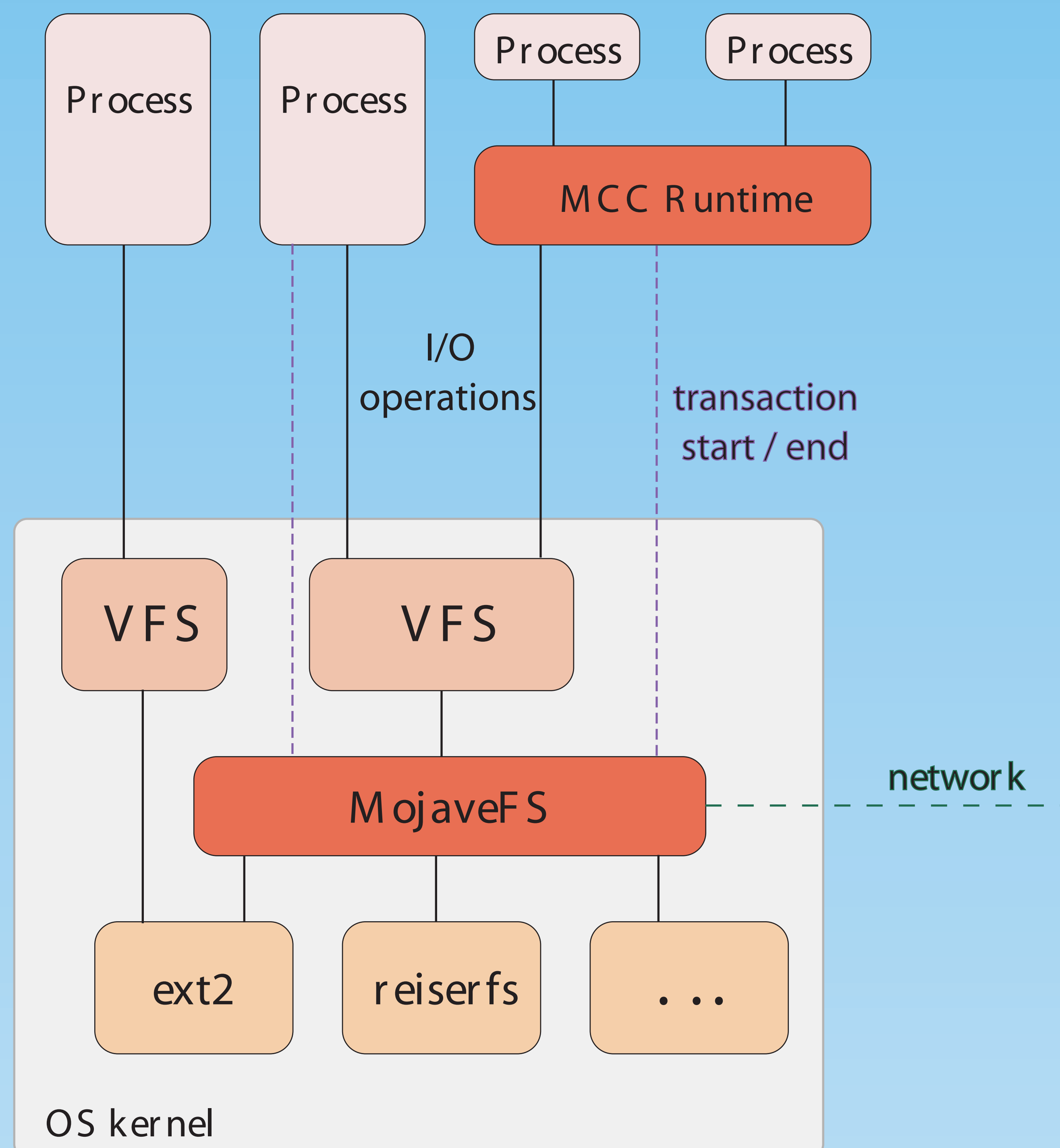
# Mojave File-System (MojaveFS)

Cristian Tăpuș, Jason Frantz, Justin Smith, Jason Hickey  
Computer Science Department, California Institute of Technology

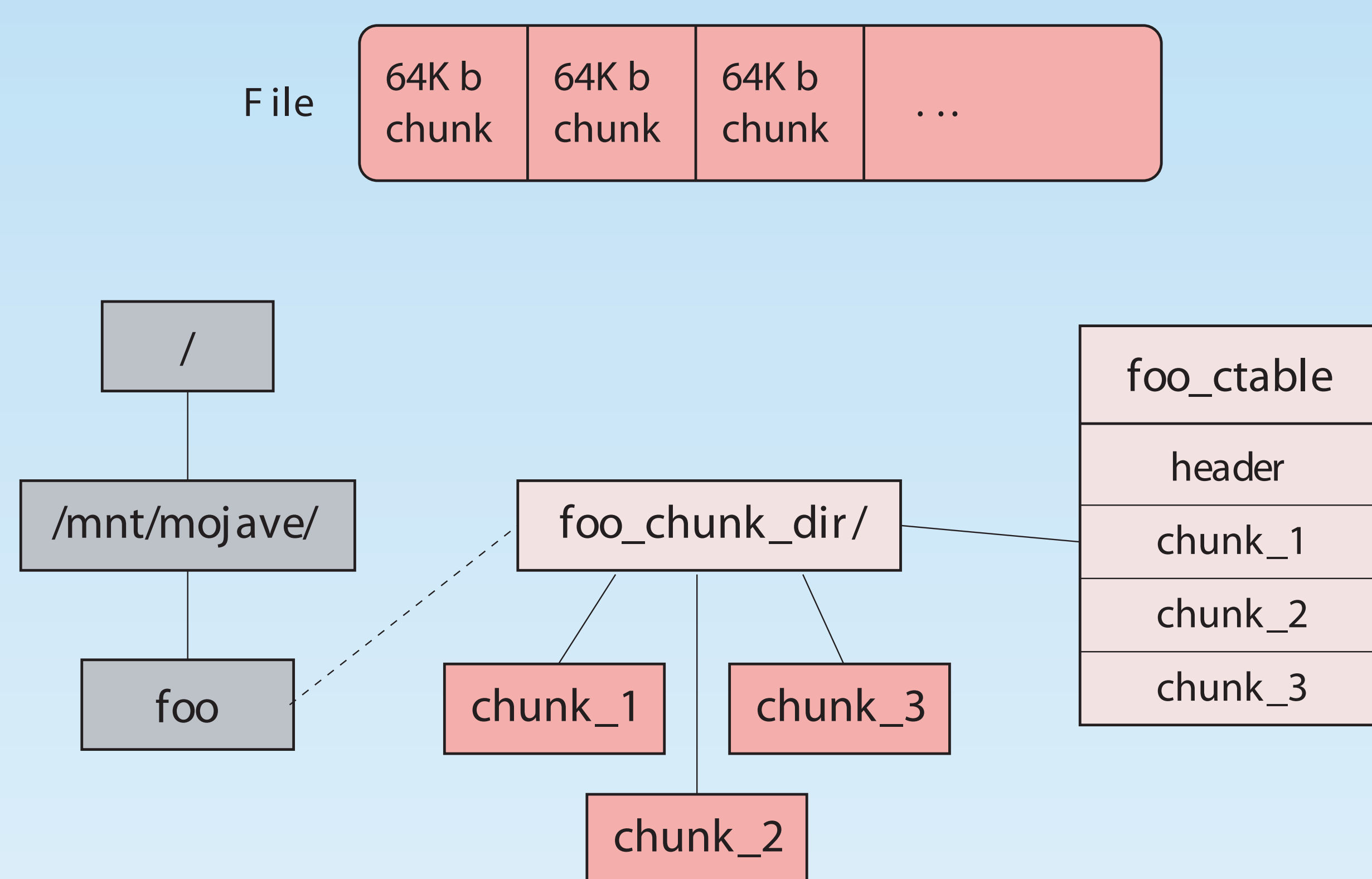
# Mojave

## Mojave File System (MojaveFS) Design

MojaveFS is a transactional, fault-tolerant distributed file system. The design goals were to make it scalable, reliable, transparent, and able to provide a global namespace.



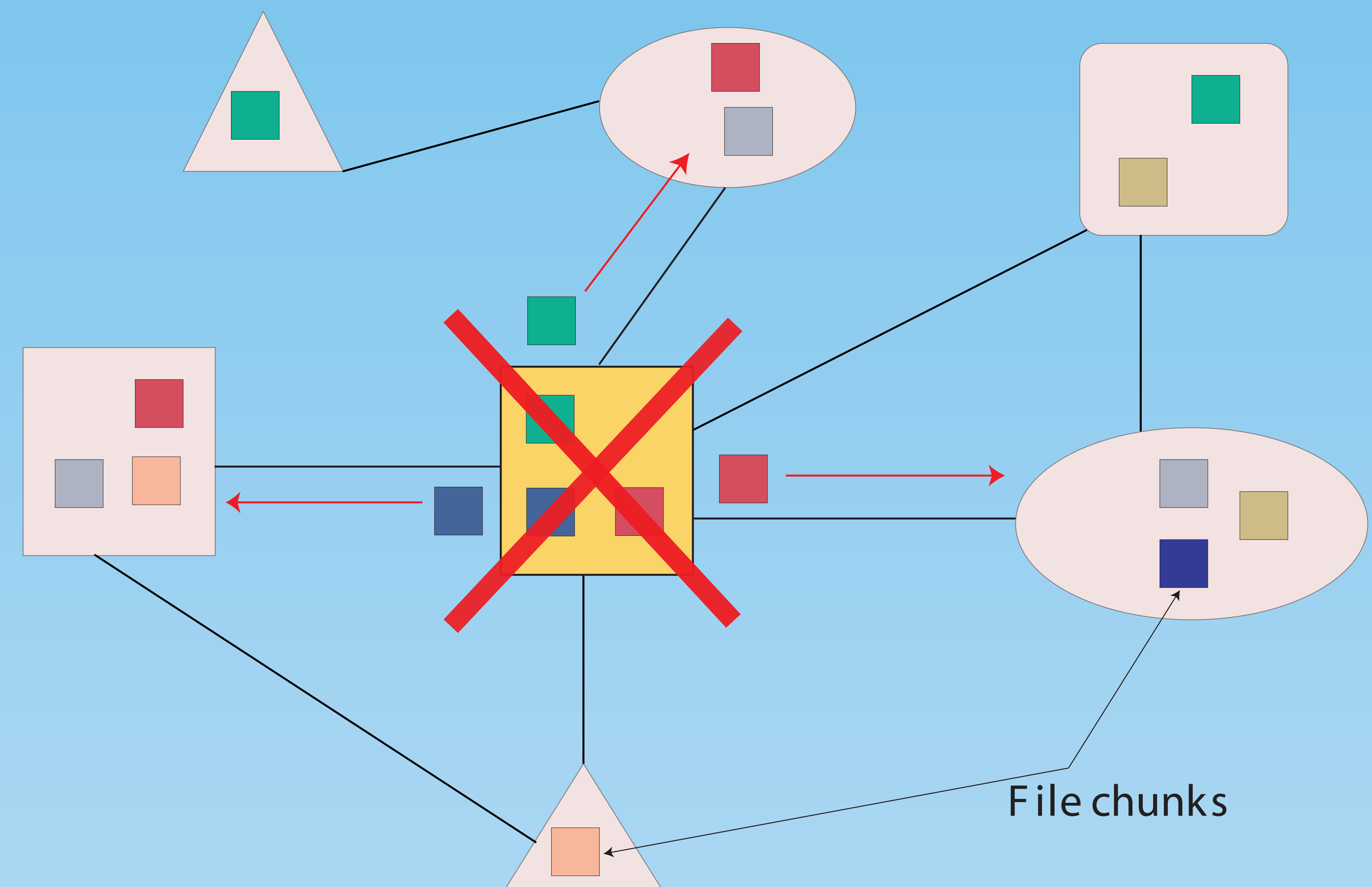
**Architecture.** MojaveFS was designed as a layer between the Virtual File System (VFS) and native local file systems. We used Linux Kernel v.2.4.18 for our implementation.



MojaveFS adopts a novel data storage strategy. Files are split into smaller "chunks" which increases data availability and decreases the access time. It also makes it easier to replicate data across the system.

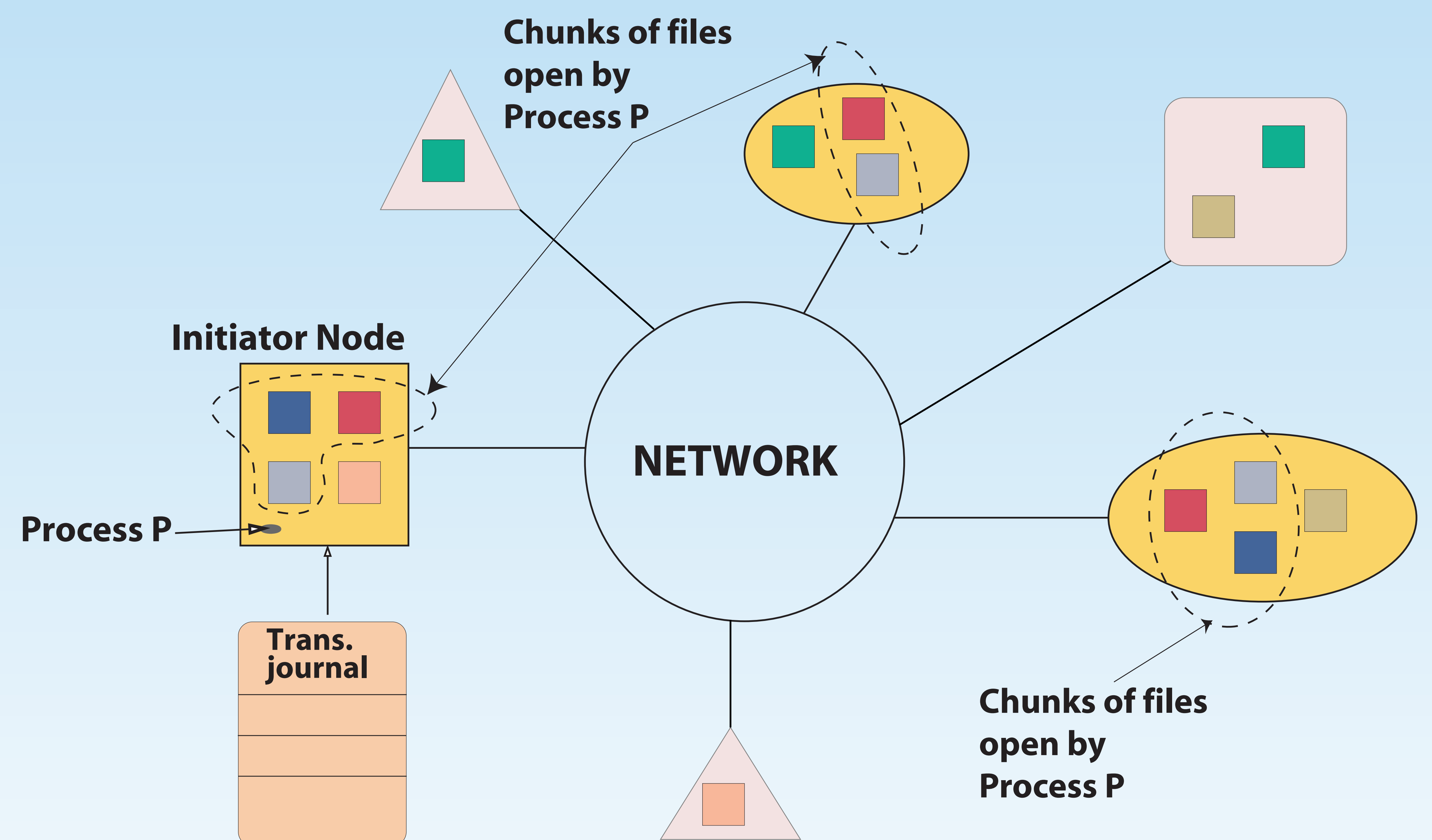
**The global namespace** means that all the nodes in the system have the same view of the file system. This feature assures the file's location transparency and makes MojaveFS a suitable infrastructure for process migration. When a process is migrated from one machine to another its view of the file system doesn't change.

**Scalability** is achieved by making the system serverless and by reducing the overhead of new machines joining the system.



**Reliability** is accomplished by data replication. When one of the nodes fails, the lost chunks are replicated again to maintain high reliability.

**Transactions** are abstractions for reliable concurrent programming. They allow processes in a distributed environment to roll back the execution to a previous valid state if a failure occurs. MojaveFS provides transactional support by allowing any file operation to be rolled back if it was part of a transaction. Along with process migration, transactions are great tools for providing fault-tolerance.



**Transactions in MojaveFS.** Process P starts a transaction. The node hosting P requests locks for the chunks of files open by P. A transaction log is created. At the end of the transaction, changes are propagated to all copies of the chunks in the system and the locks are released.