

# SeFS

Simple, Scalable, Flash-Optimized and



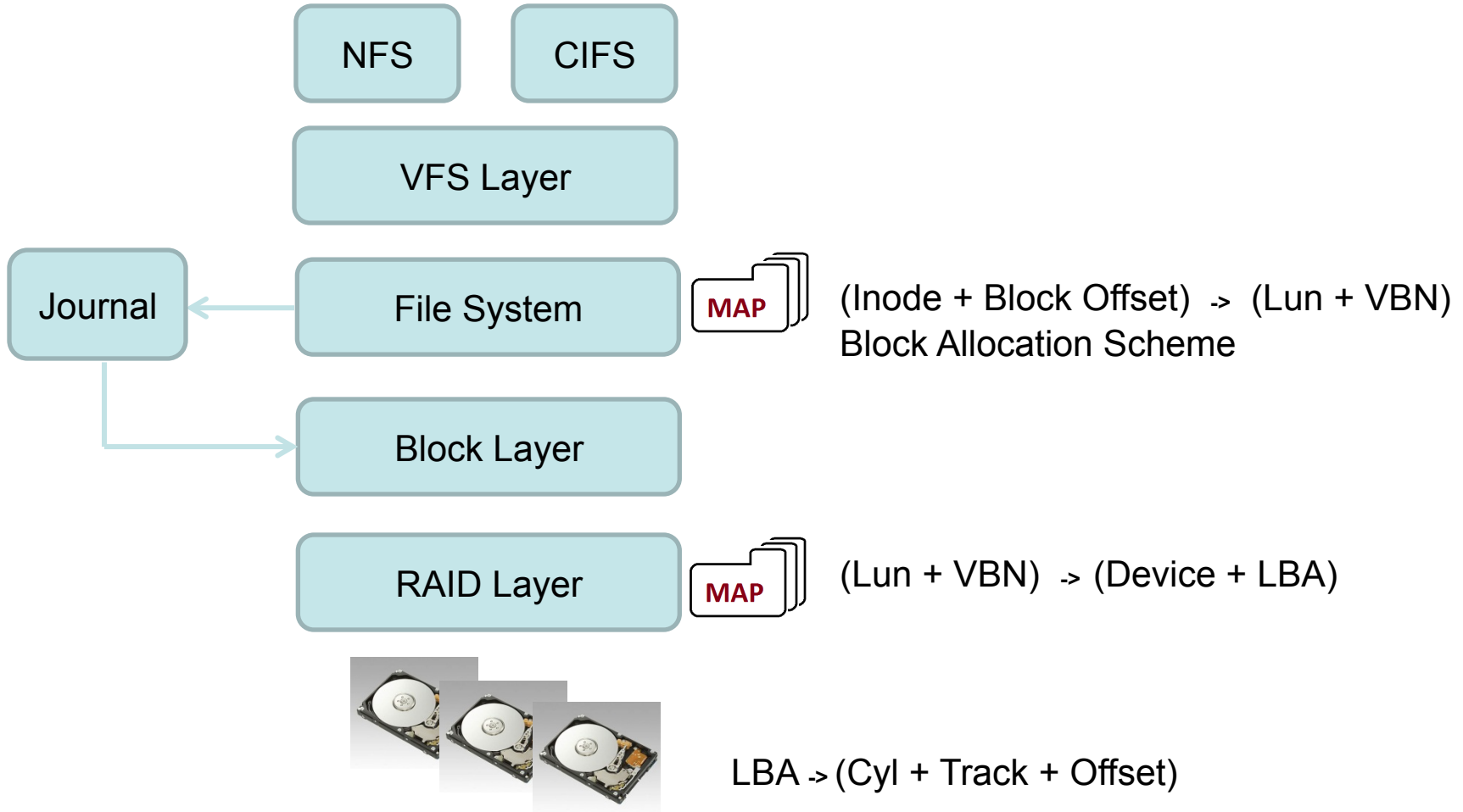
Amit Bothra  
Arvind Pruthi



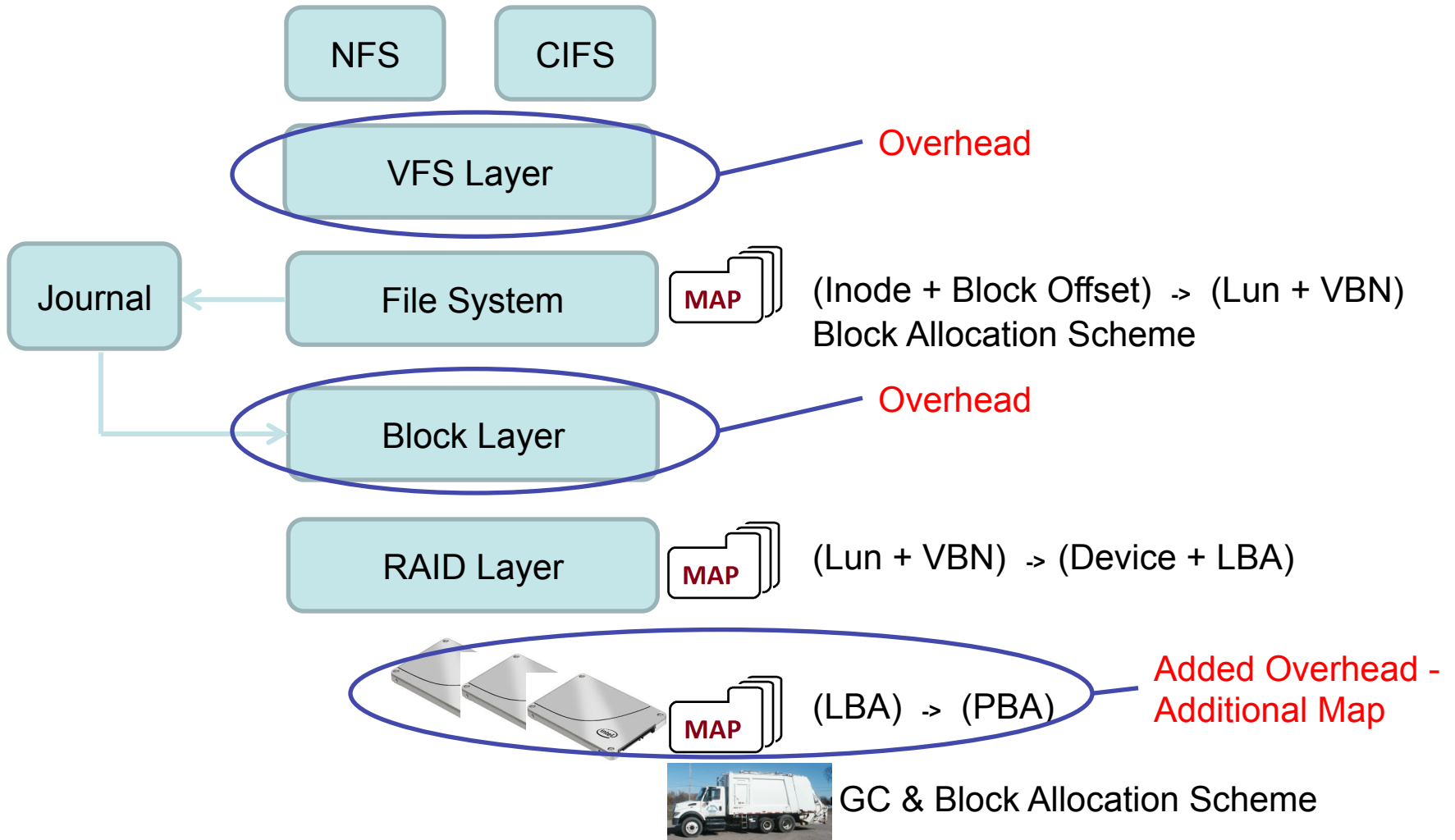
# Efficiency at Every Level

- Reads
- Writes
- Journaling
- Block Allocation
- Garbage Collection
- Low Memory Overhead
- Simple, yet scalable

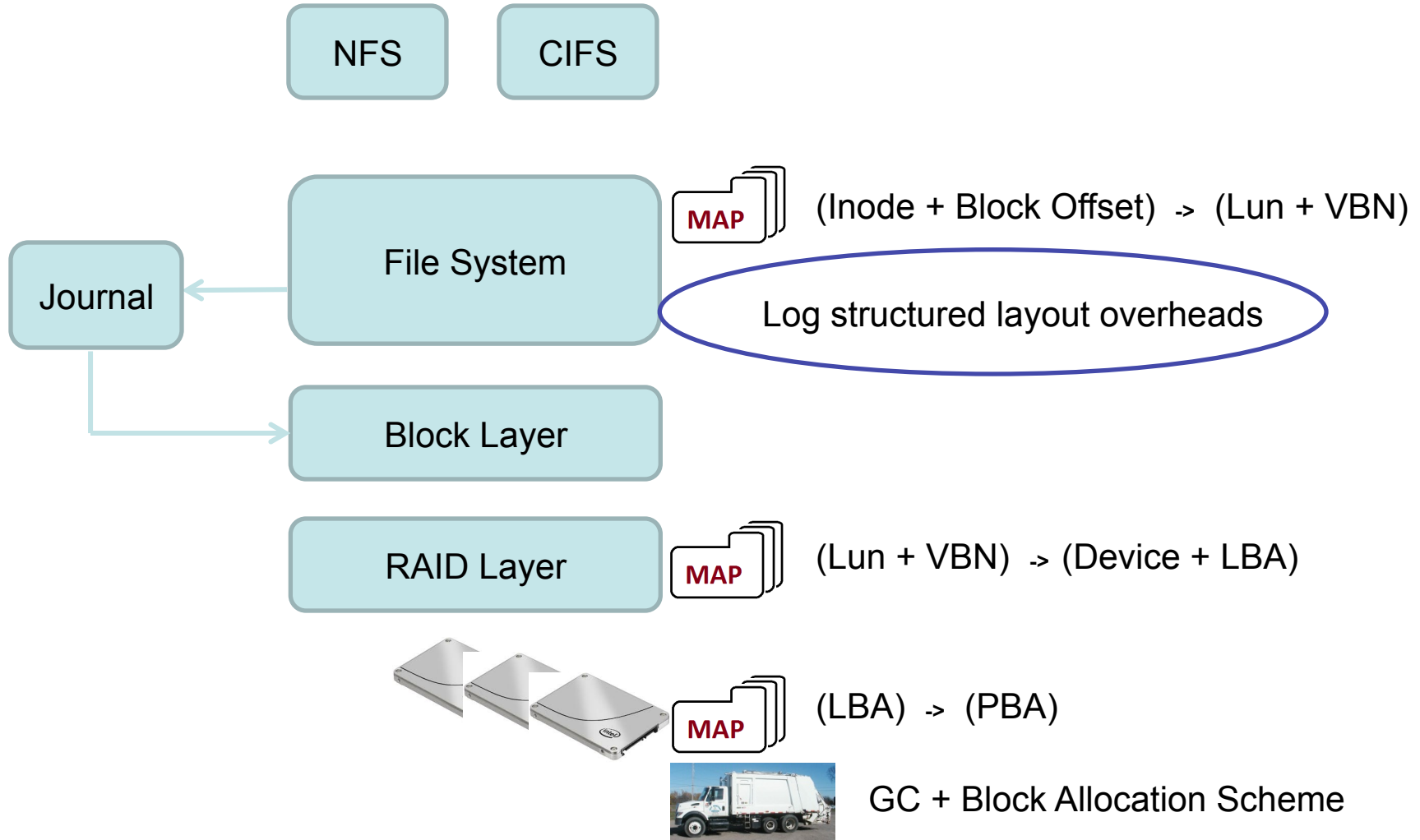
# Traditional File System Stack



# Traditional File System Stack w/ SSDs



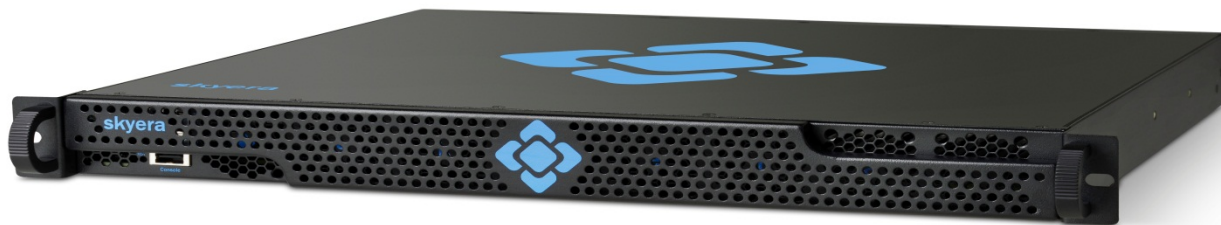
# Enterprise Log-Structured File System Stack w/SSDs



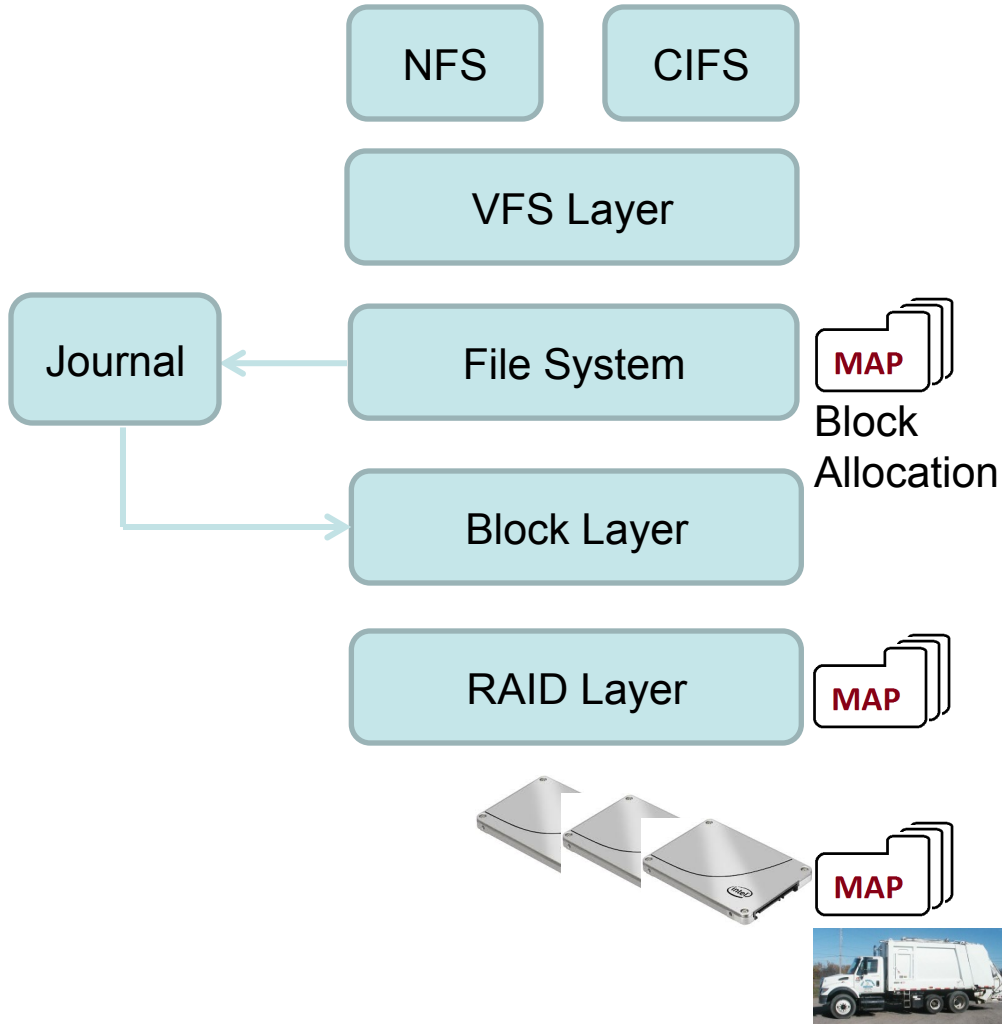


# Enter Skyhawk

Vertically  
Integrated  
File System  
Stack



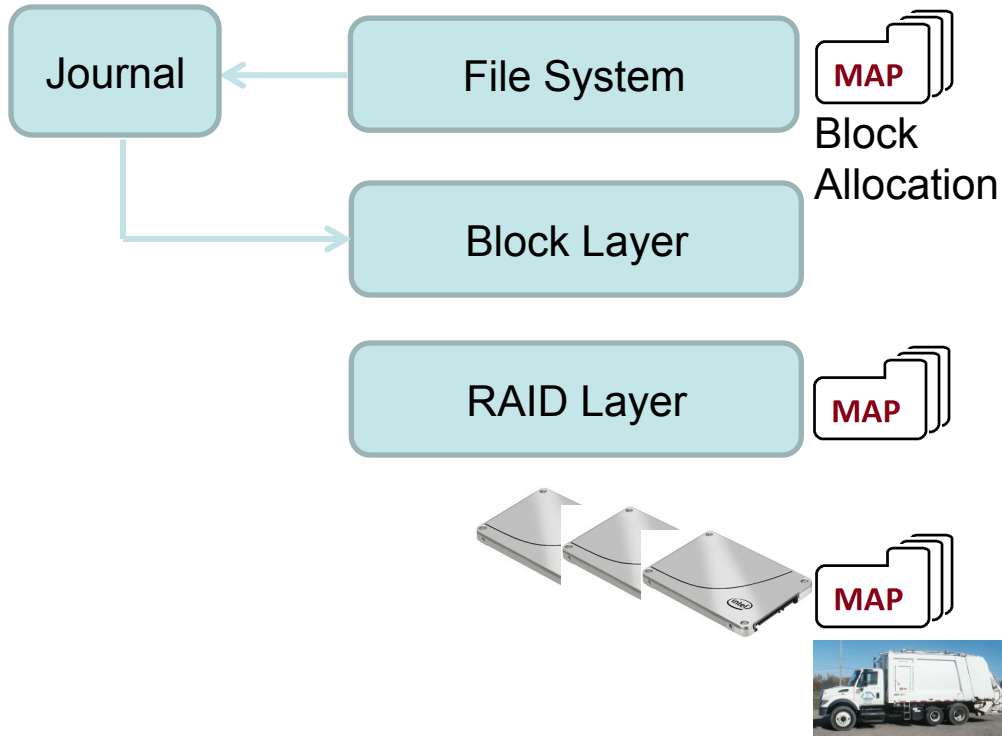
# Benefits of SeFS



# Benefits of SeFS

NFS

CIFS

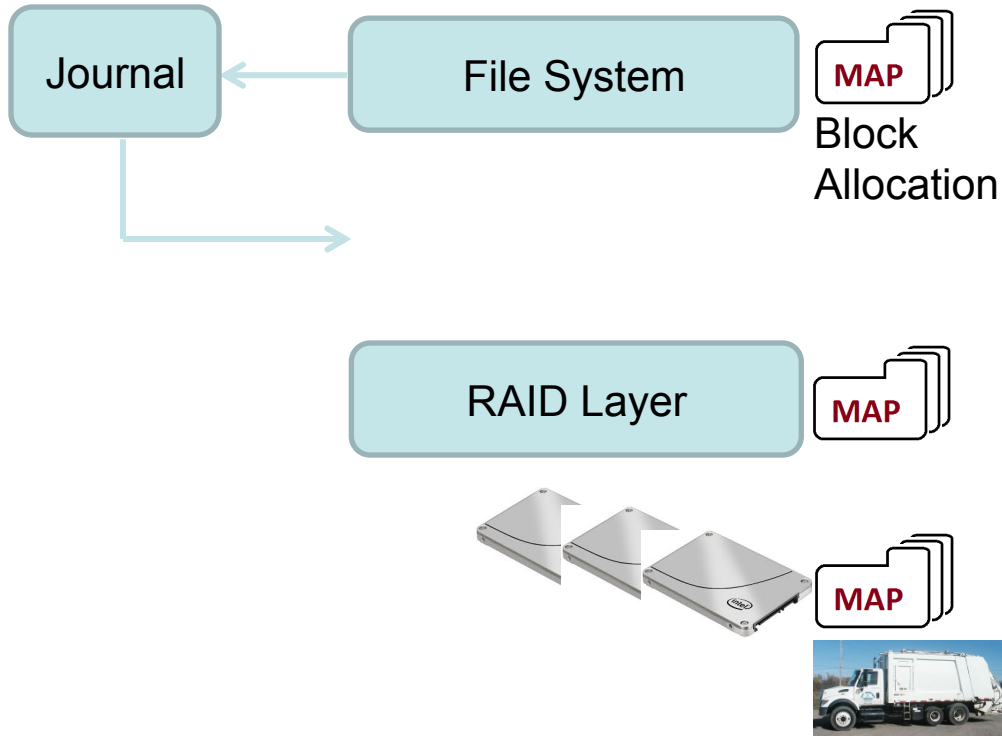




# Benefits of SeFS

NFS

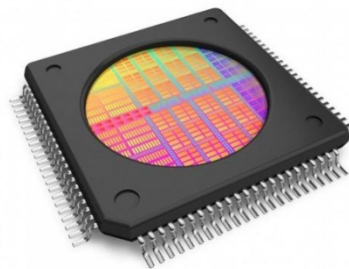
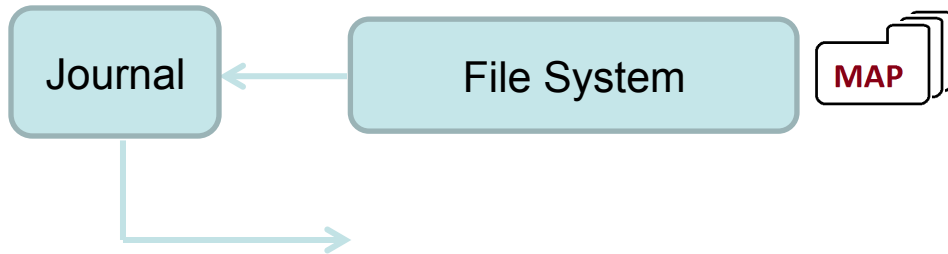
CIFS



# Benefits of SeFS

NFS

CIFS



- FTL with Global wear Levelling
- Sparse VBN -> PBA Mapping
- Block Allocation
- GC

# Benefits of SeFS

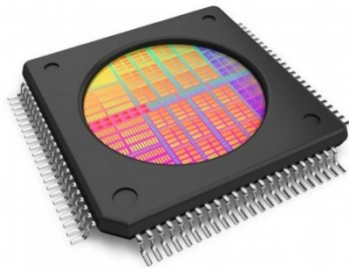
NFS

CIFS

Skyera FS

## Zero Mapping

- Every FS Object has its own Virtual Block Space
- No mapping required at FS
- All block allocation is done by FTL
- No journaling needed by File System



- FTL with Global wear Levelling
- Sparse VBN -> PBA Mapping
- Block Allocation
- GC



## To Emphasize

- Read-path is very efficient with a single mapping translation
- Write path is very efficient with a single mapping translation, no block allocation and no journaling overhead
- File system architecture designed to keep wear low on flash by low metadata overhead.
- Over-Provisioning of storage strictly limited to flash.
- All the above is possible only with a ground-up re-architecture with vertically integrated storage stack!