Advanced Programming in the UNIX Environment

Week 04, Segment 1: The Unix Filesystem

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

| hard disk | partition | partition | partition |

- a disk can be divided into logical *partitions*

Jan Schaumann   2020-09-17

```
Terminal — 80×27

[apue$ sudo disklabel wd0
# /dev/rwd0:
type: unknown
disk: wd
label: fictious
flags:
bytes/sector: 512                        physical blocksize
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 33288
total sectors: 33554432
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0              # microseconds
track-to-track seek: 0  # microseconds
drivedata: 0


4 partitions:
#        size     offset     fstype [fsize bsize cpg/sgs]
 a:   31457216        64     4.2BSD       0      0     0  #     first NetBSD partition
 b:    2097088  31457344       swap                     #     second NetBSD partition
 c:   33554368        64     unused       0      0        #     NetBSD portion of disk
 d:   33554432         0     unused       0      0        #     entire disk
apue$
```

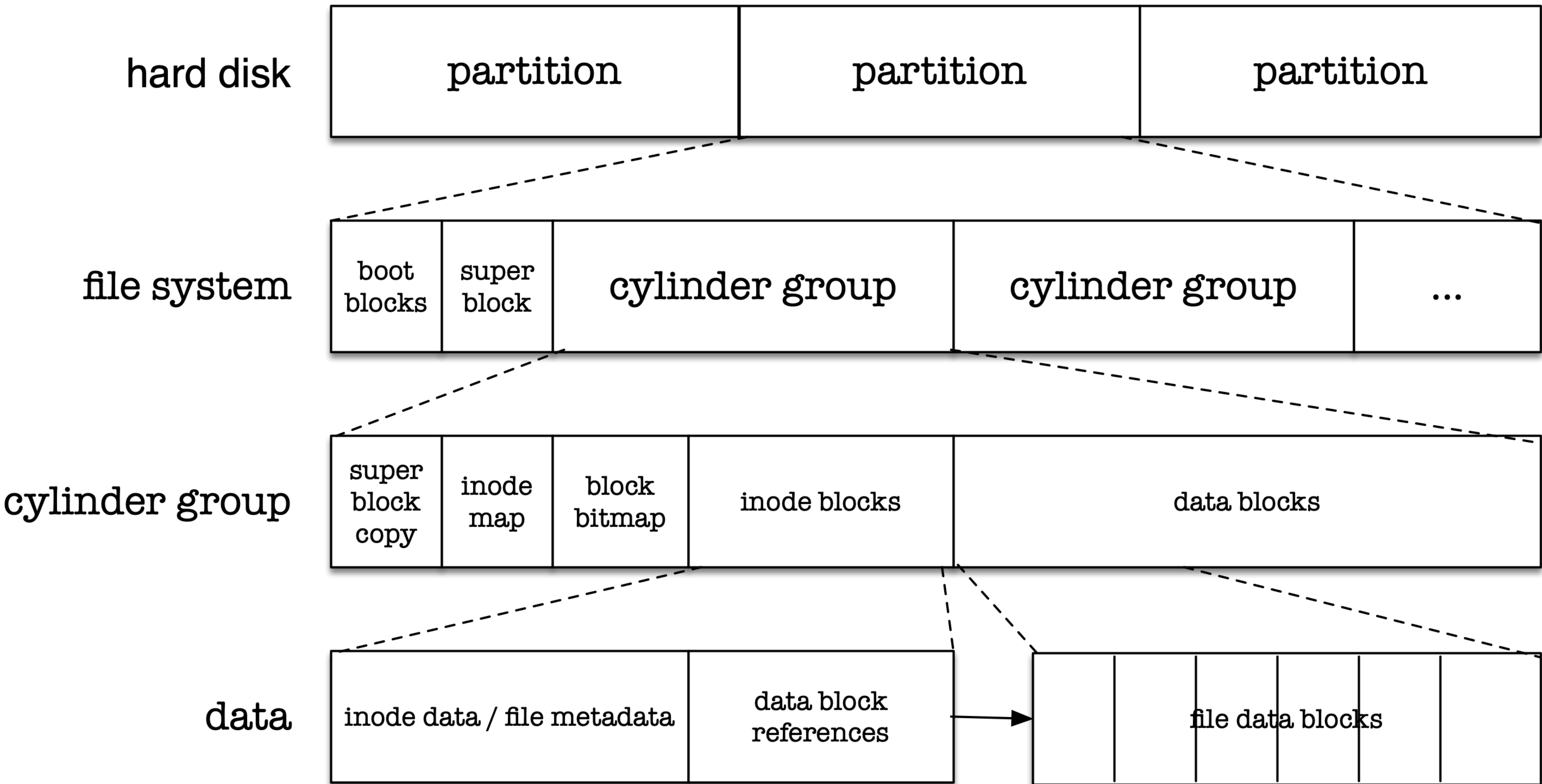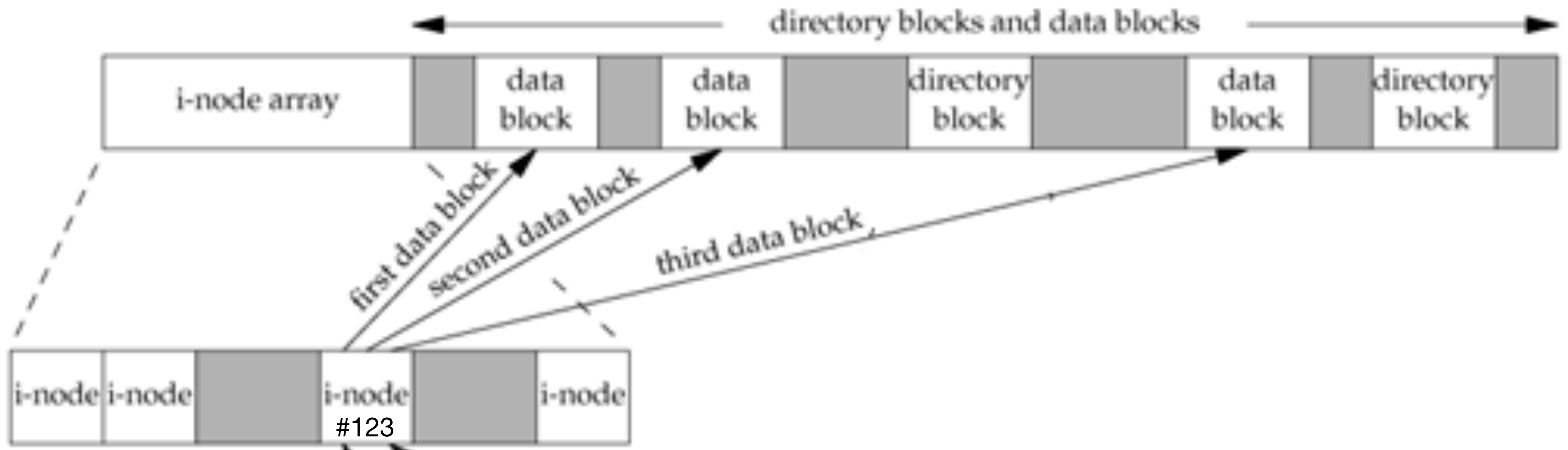- on each logical *partition* you may create a *file system* containing the cylinder groups

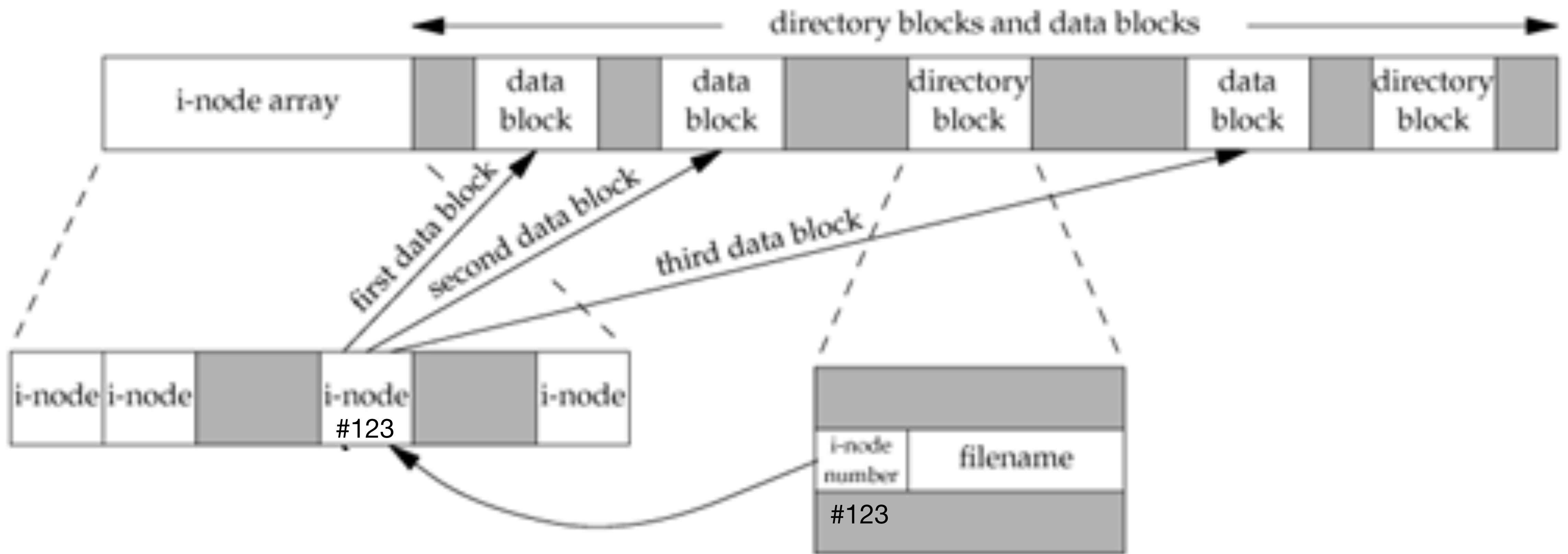Jan Schaumann   2020-09-17
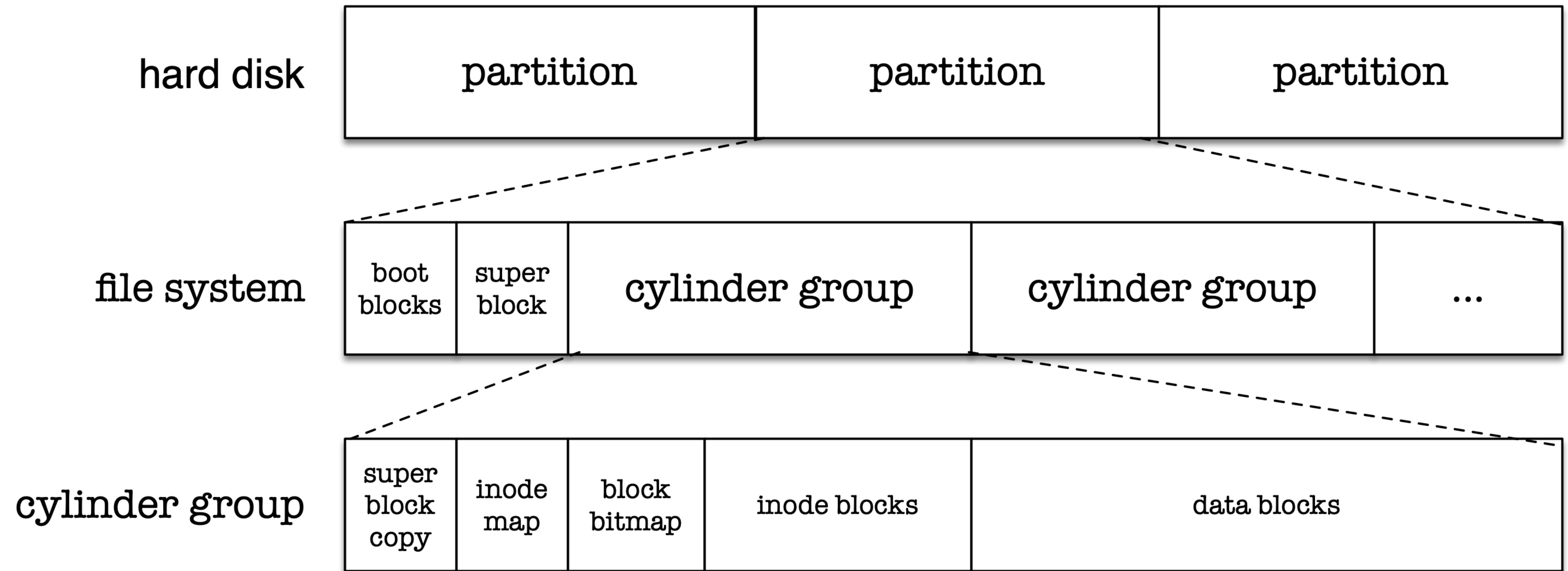
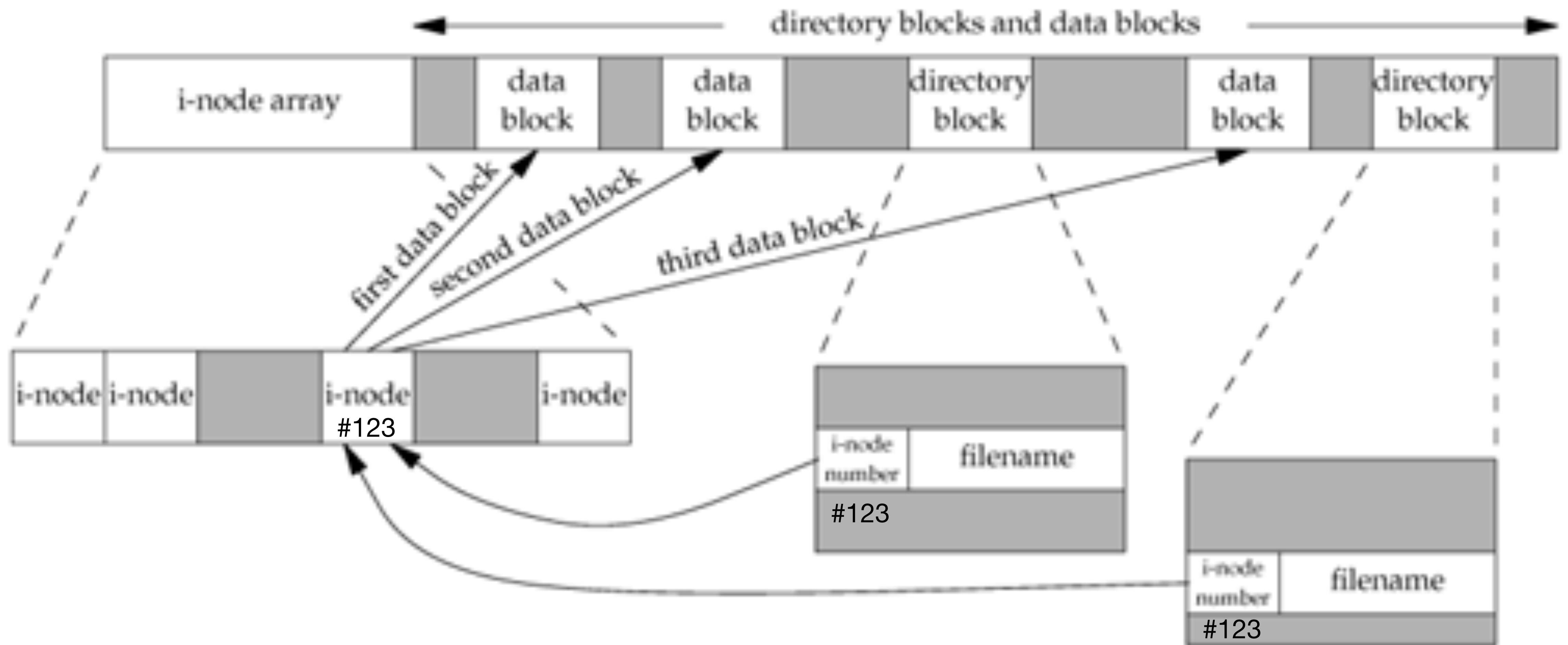- each *cylinder group* contains a list of *inodes* (i-list) as well as the actual *directory-* and *data* blocks

Jan Schaumann   2020-09-17

| hard disk | partition | partition | partition |

file system

| boot blocks | super block | cylinder group | cylinder group | ... |

cylinder group

| super block copy | inode map | block bitmap | inode blocks | data blocks |

data

| inode data / file metadata | data block references | → | file data blocks |

Jan Schaumann   2020-09-17

- data blocks containing the actual data (*i.e.*, contents of the file) are referenced from the inode

directory blocks and data blocks

i-node array | data block | data block | directory block | data block | directory block

first data block
second data block
third data block

i-node | i-node | i-node #123 | i-node
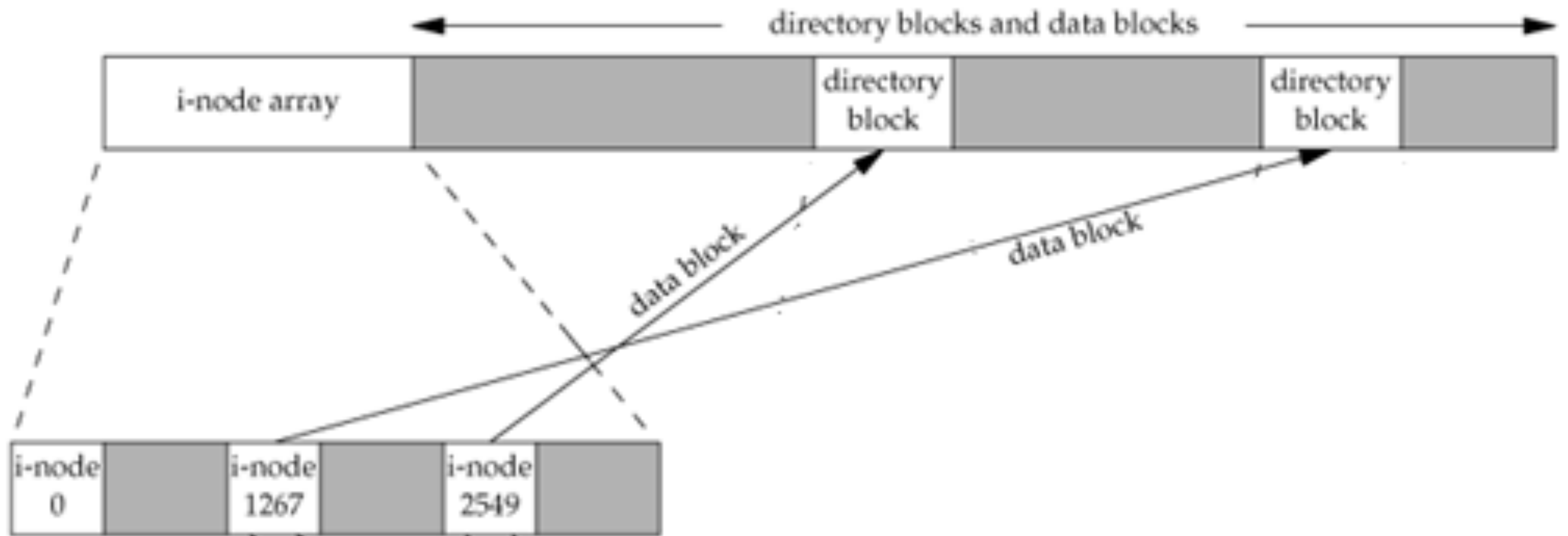
i-node number | filename

#123

#123

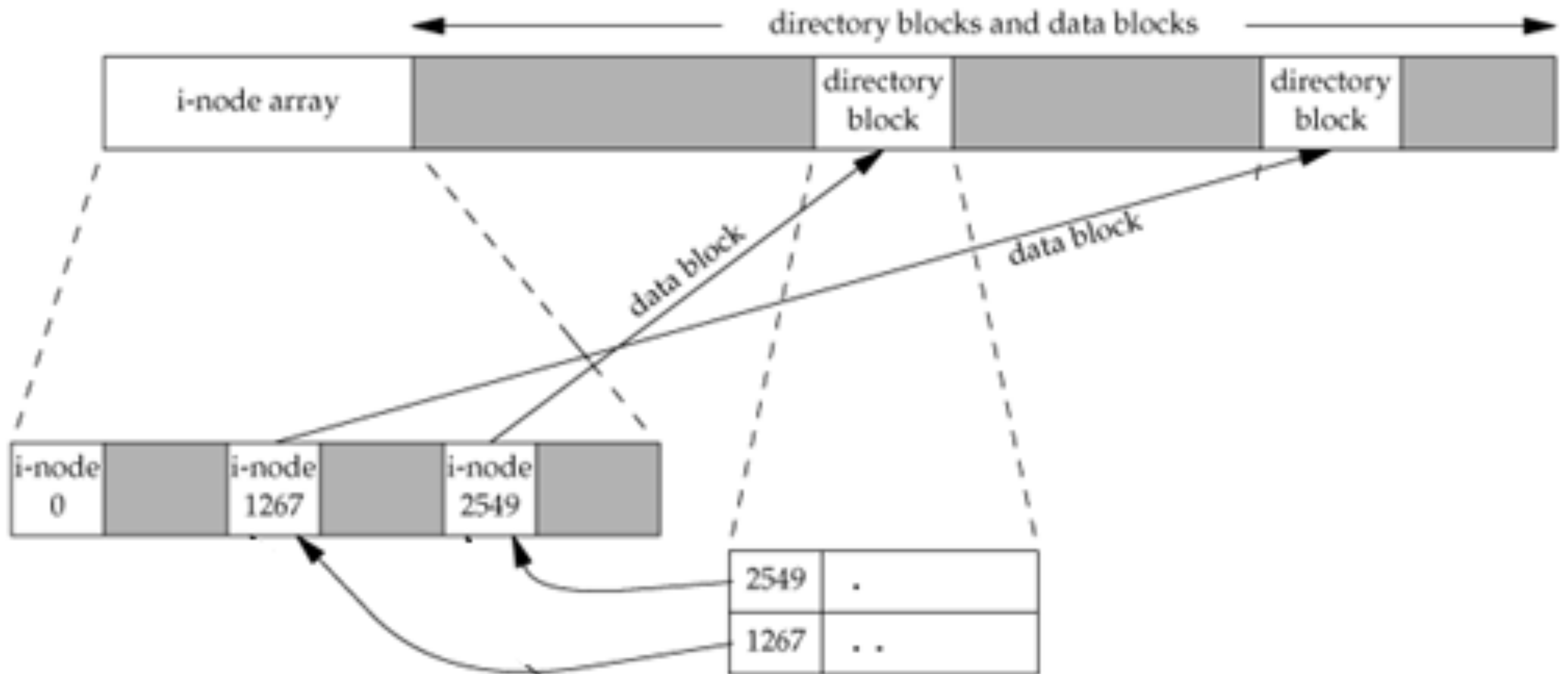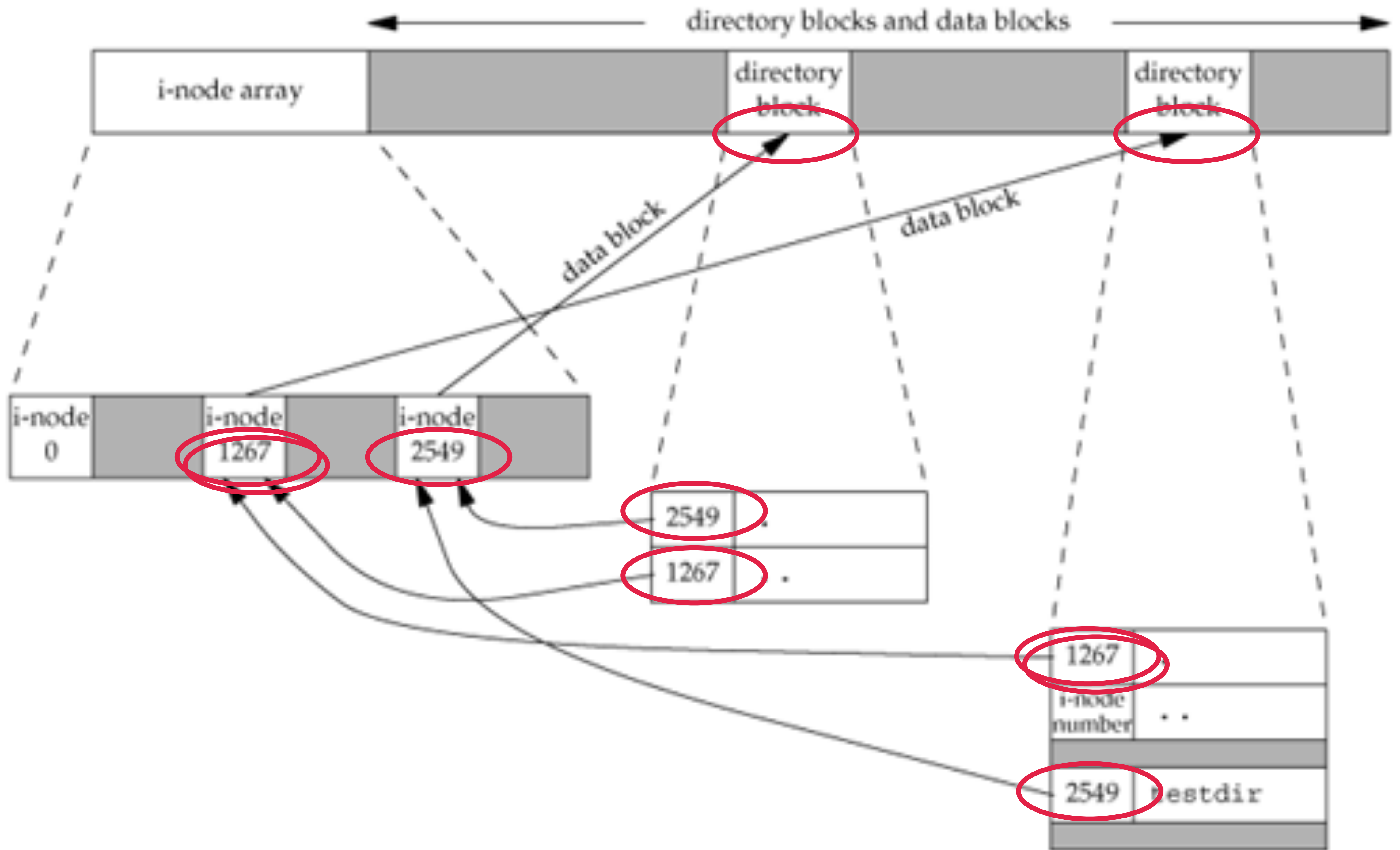- a directory entry is really just a *hard link* mapping a "filename" to an inode

- a directory entry is really just a *hard link* mapping a "filename" to an inode

- you can have many such mappings to the same inode

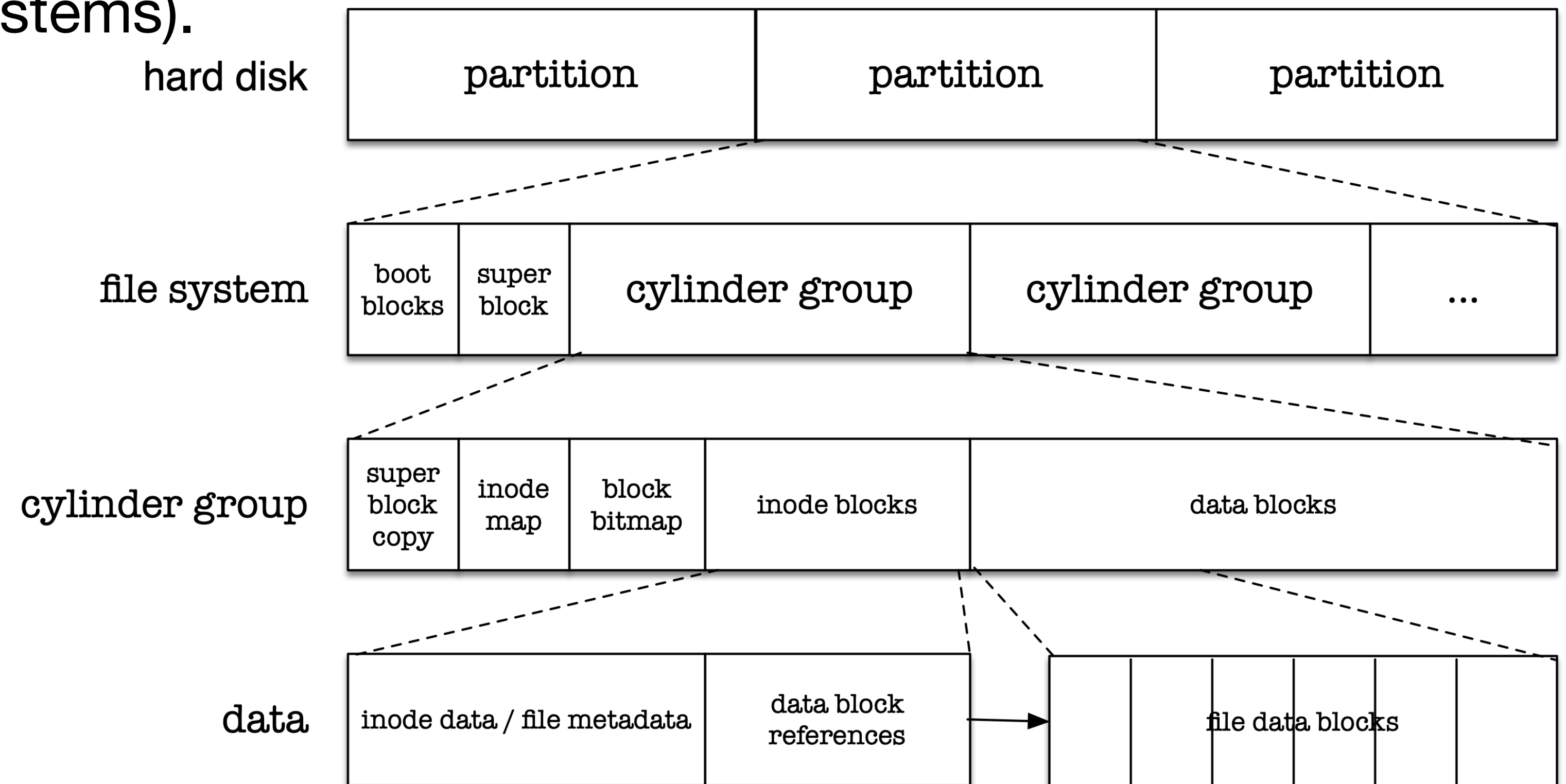- directories are special "files" containing a list of hard links

- each directory contains at least two entries:

  - "." -- this directory

  - ".." -- the parent directory

directory blocks and data blocks

i-node array

directory block

directory block

data block

data block

i-node 0

i-node 1267

i-node 2549

| 2549 | . |
| 1267 | .. |

1267	
i-node number	..
2549	testdir

Inodes

- The inode number in a directory entry must point to an inode on the same file system (no hardlinks across filesystems).

hard disk	partition	partition	partition

file system	boot blocks	super block	cylinder group	cylinder group	...

cylinder group	super block copy	inode map	block bitmap	inode blocks	data blocks

data	inode data / file metadata	data block references	file data blocks

Jan Schaumann   2020-09-17

Inodes

- The inode number in a directory entry must point to an inode on the same file system (no hardlinks across filesystems).

- The inode contains most of the information found in the `struct stat`.

- Every inode has a link count (`st_nlink`): it shows how many "things" point to this inode. Only if this link count is 0 (and no process has the file open) are the data blocks freed.
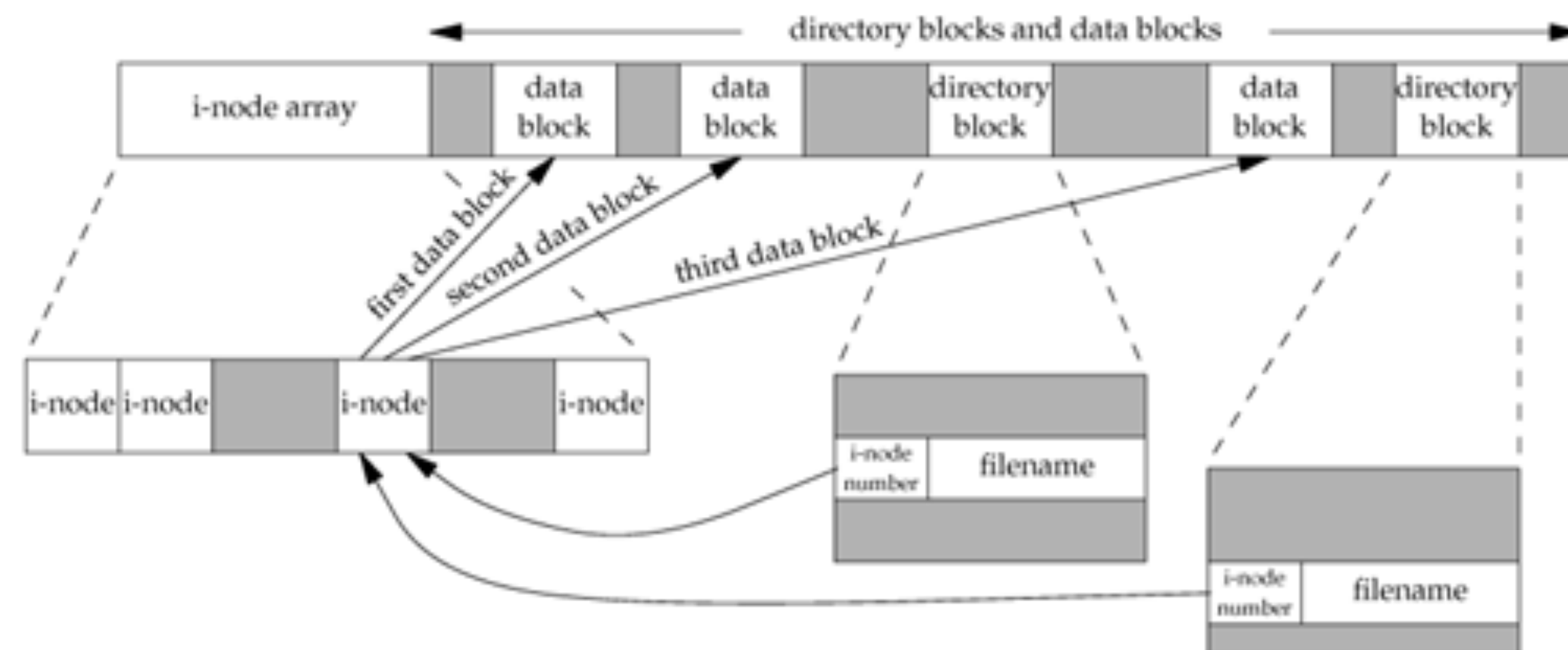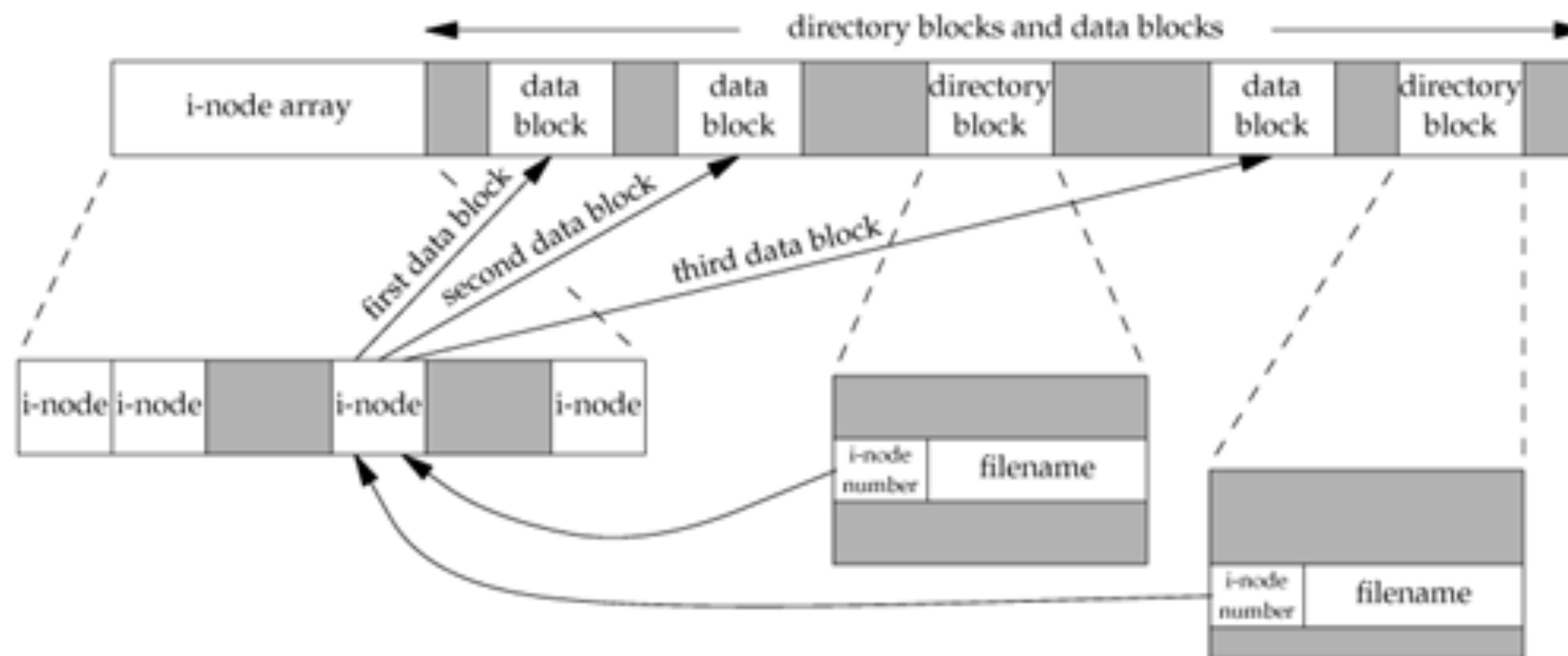
Inodes

- To move a file within a single filesystem, we can just "move" the directory entry (actually done by creating a new entry, and deleting the old one).

Jan Schaumann   2020-09-17
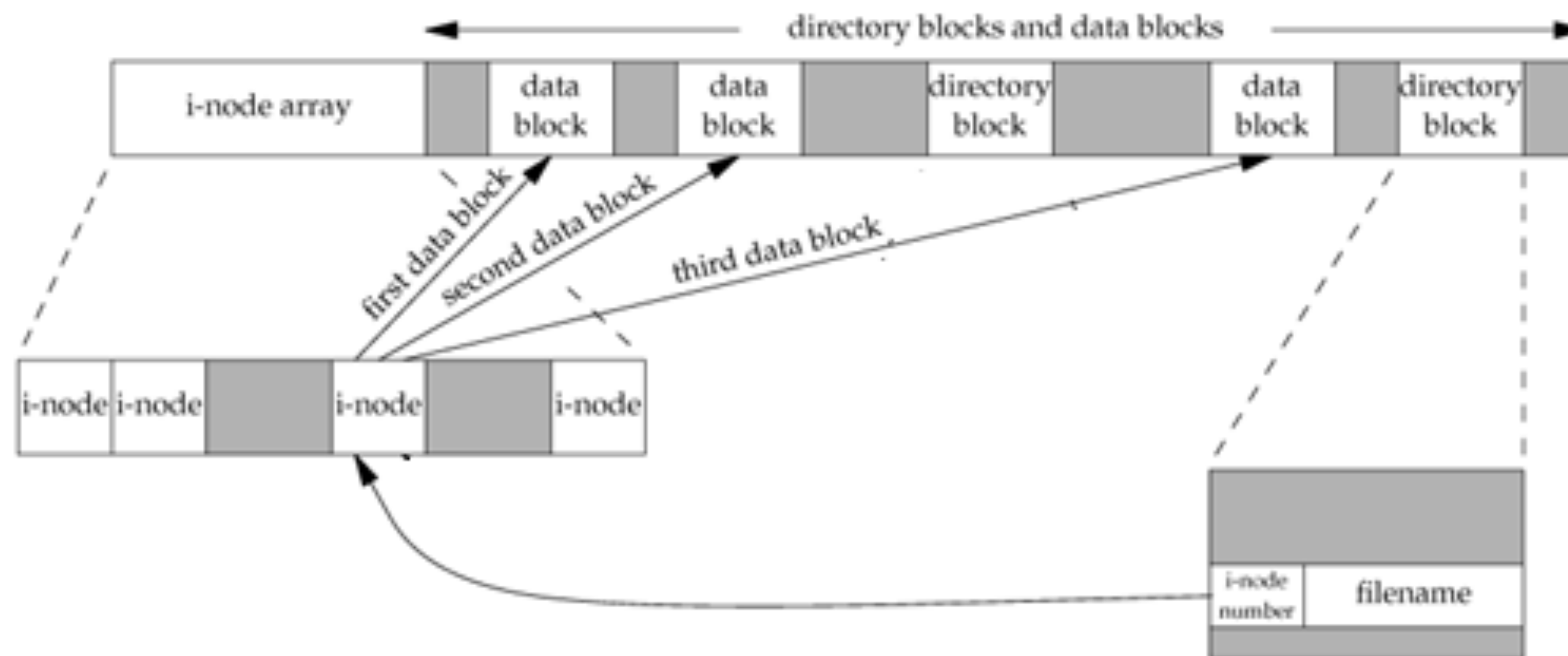
Inodes

- To move a file within a single filesystem, we can just "move" the directory entry (actually done by creating a new entry, and deleting the old one).

Jan Schaumann   2020-09-17
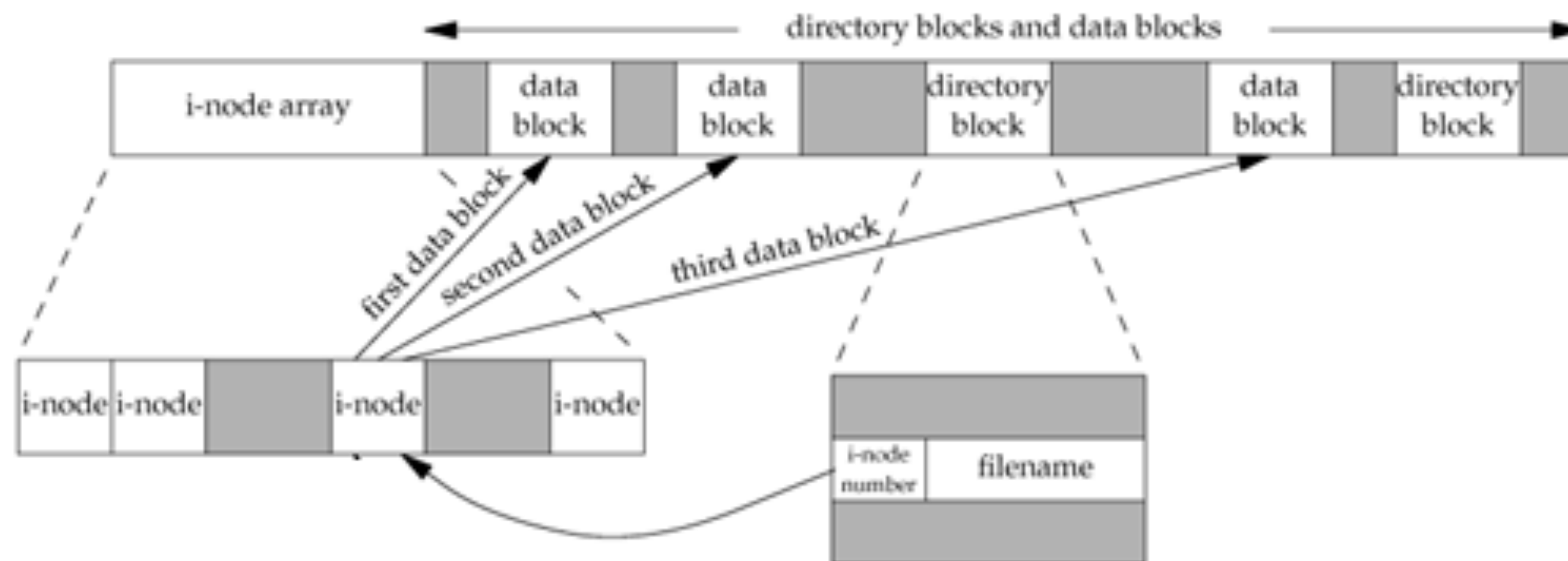
Inodes

- To move a file within a single filesystem, we can just "move" the directory entry (actually done by creating a new entry, and deleting the old one).

The Unix Filesystem

Visualizing the Unix Filesystem helps us understand the concept of *hard links*, what directories "look like", and how operations on a directory are independent of the files and their data.

Coming up: creating, removing, and renaming links (*hard* and *symbolic*)

link(2)   unlink(2)   rename(2)   symlink(2) / readlink(2)

Jan Schaumann   2020-09-17