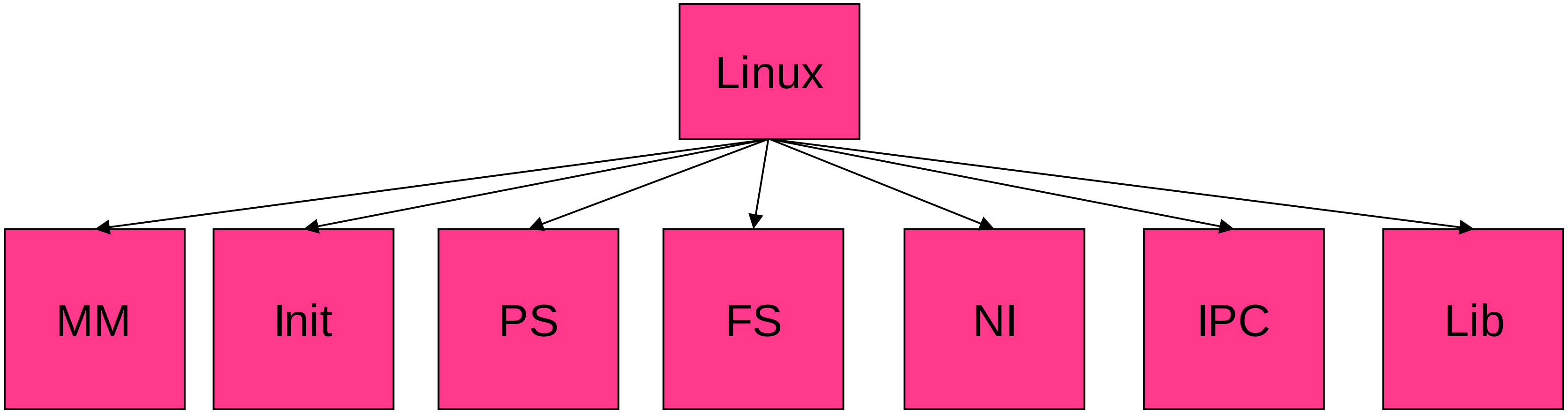
4+1 View of Architecture

What does Linux have?

- Linux subsystems
  - Process Scheduler (PS) – responsible for supporting multitasking by deciding which user process executes.
  - Memory Manager (MM) – provides a separate memory space for each user process.
  - File System (FS)– provides access to hardware devices
  - Network Interface (NI)– encapsulates access to network devices

- Linux subsystems
  - Inter Process Communication (IPC)– allows user processes to communicate with other processes on the same computer
  - Initialization (Init)– responsible for initializing the rest of the linux kernel with appropriate usr configured settings
  - Library (Lib)– the kernel core which stores the routines that are used by other subsystems for their running.

Linux



Linux

MM

Init

PS

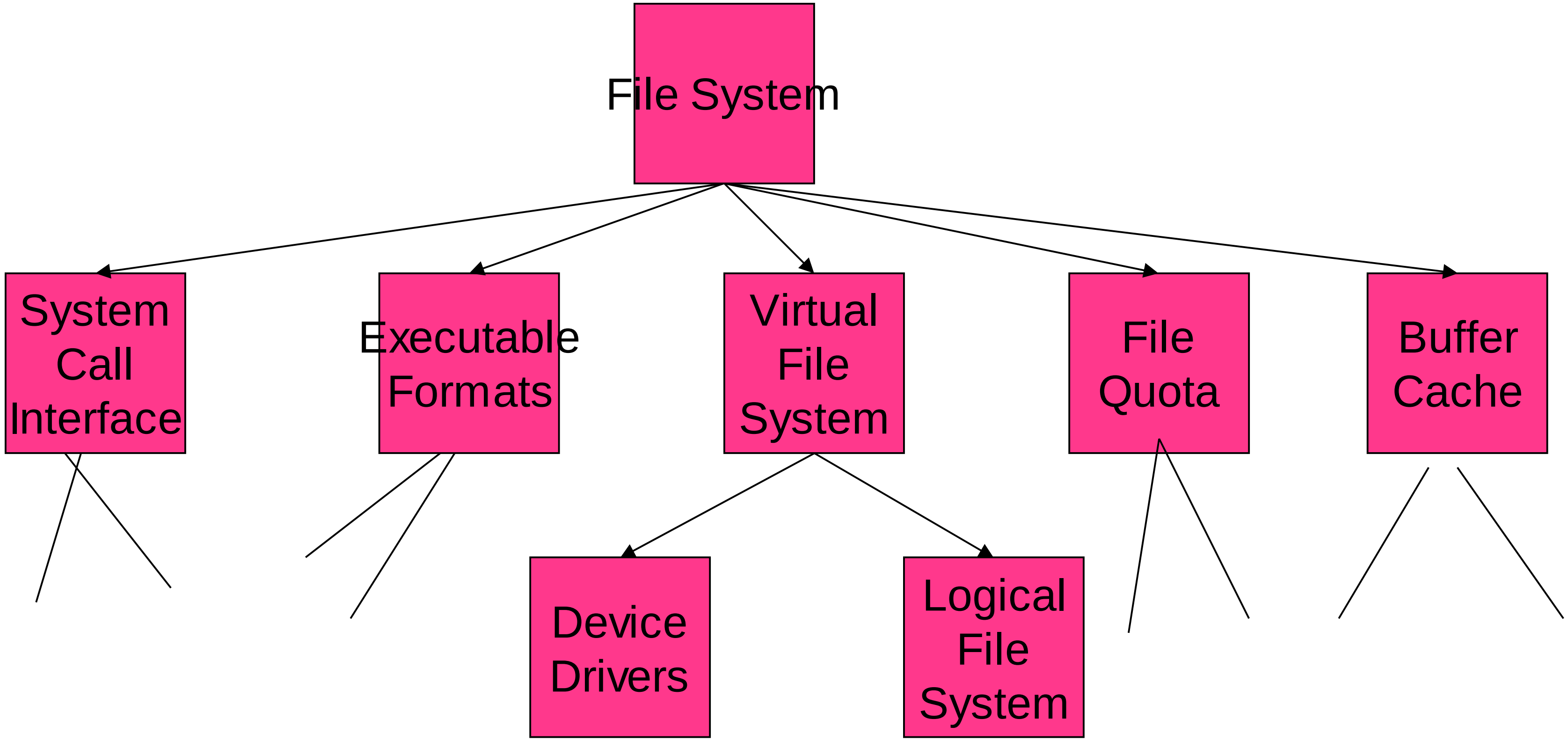
FS

NI

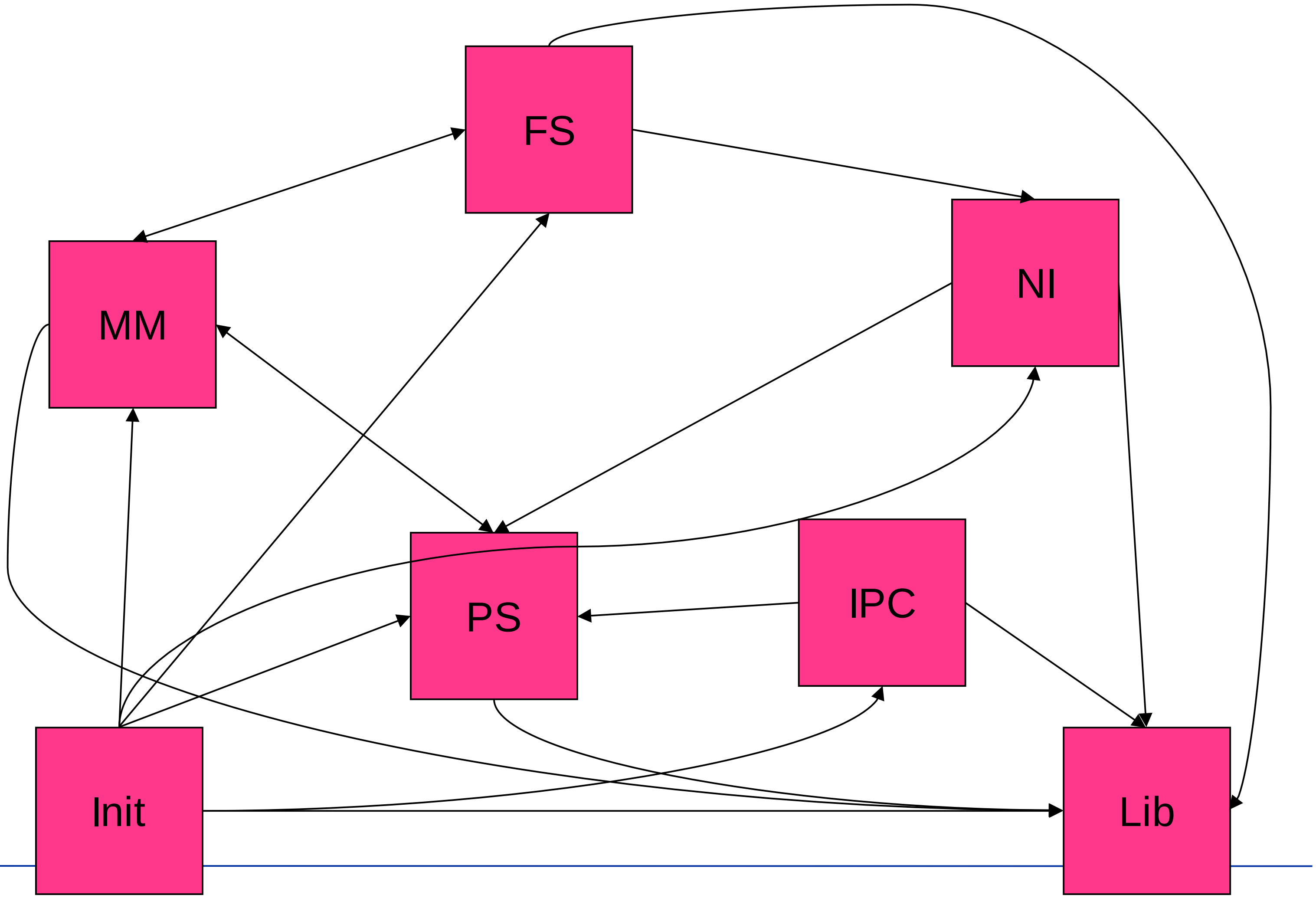
IPC

Lib

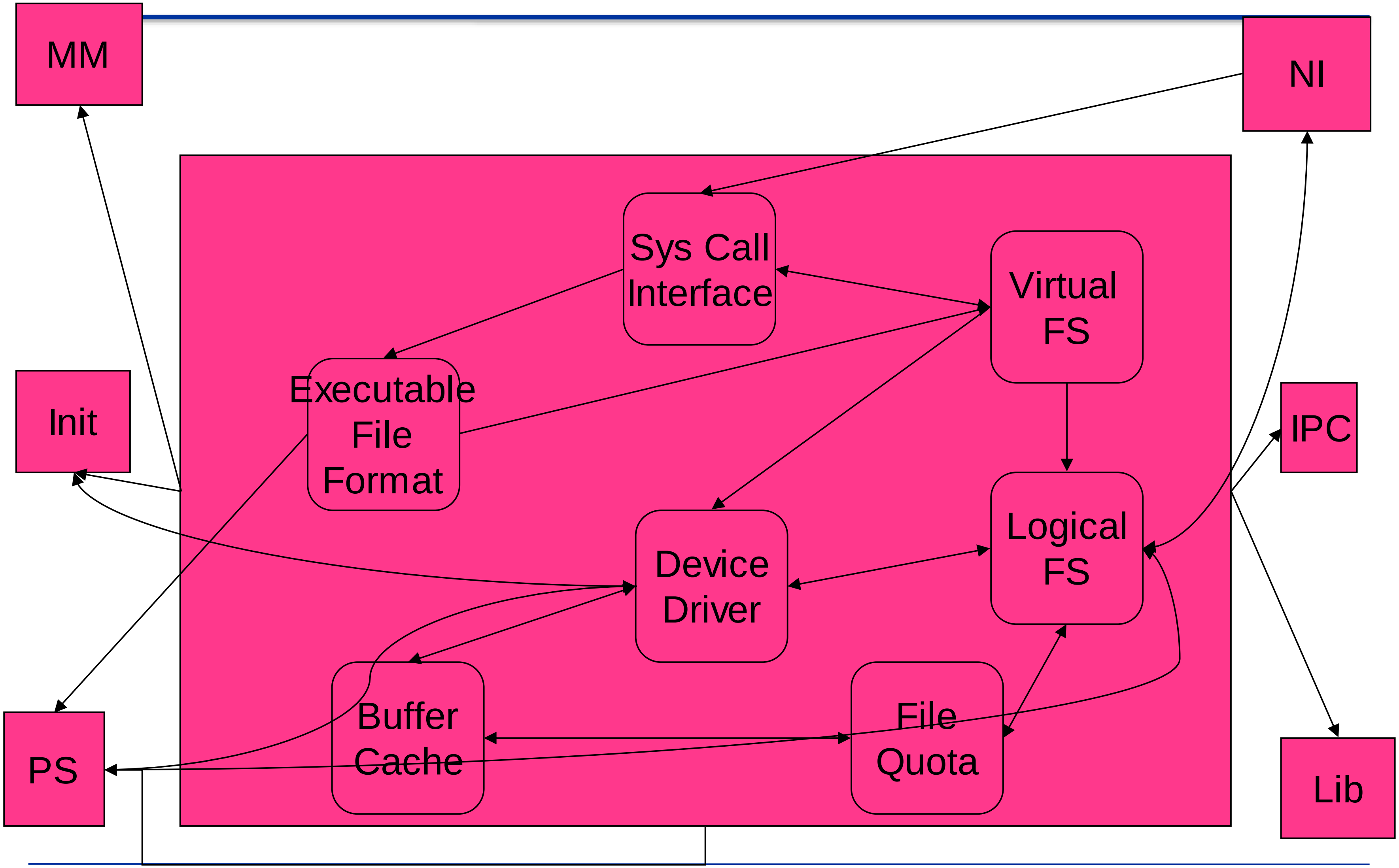
File System Broken Down



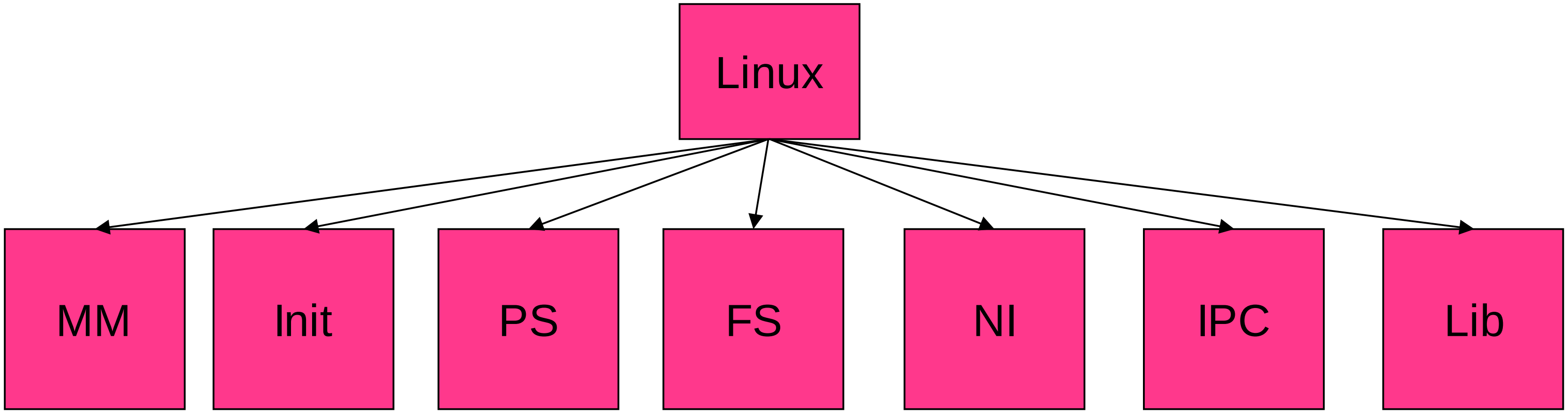
How do these sub-systems interact?



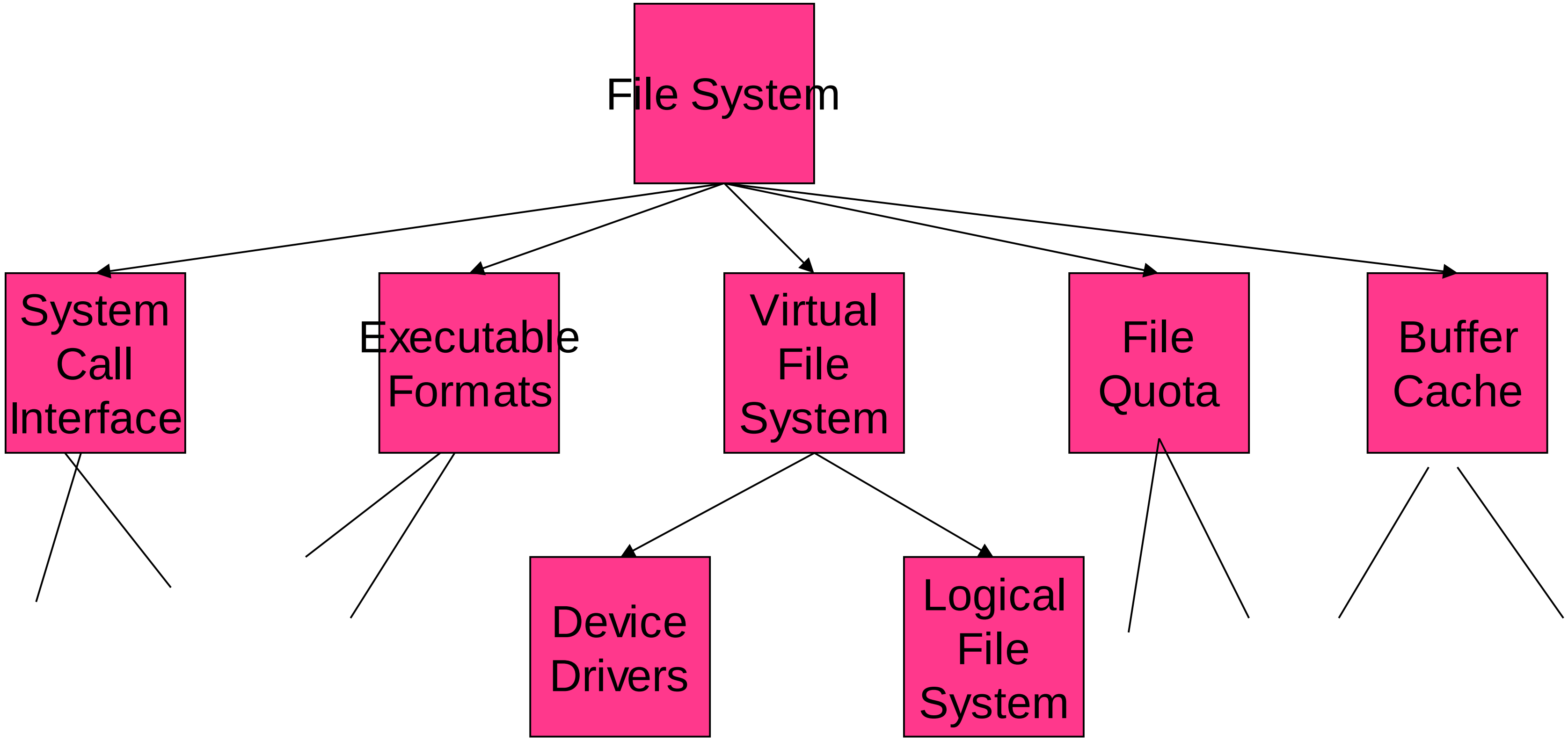
Breakdown of FS



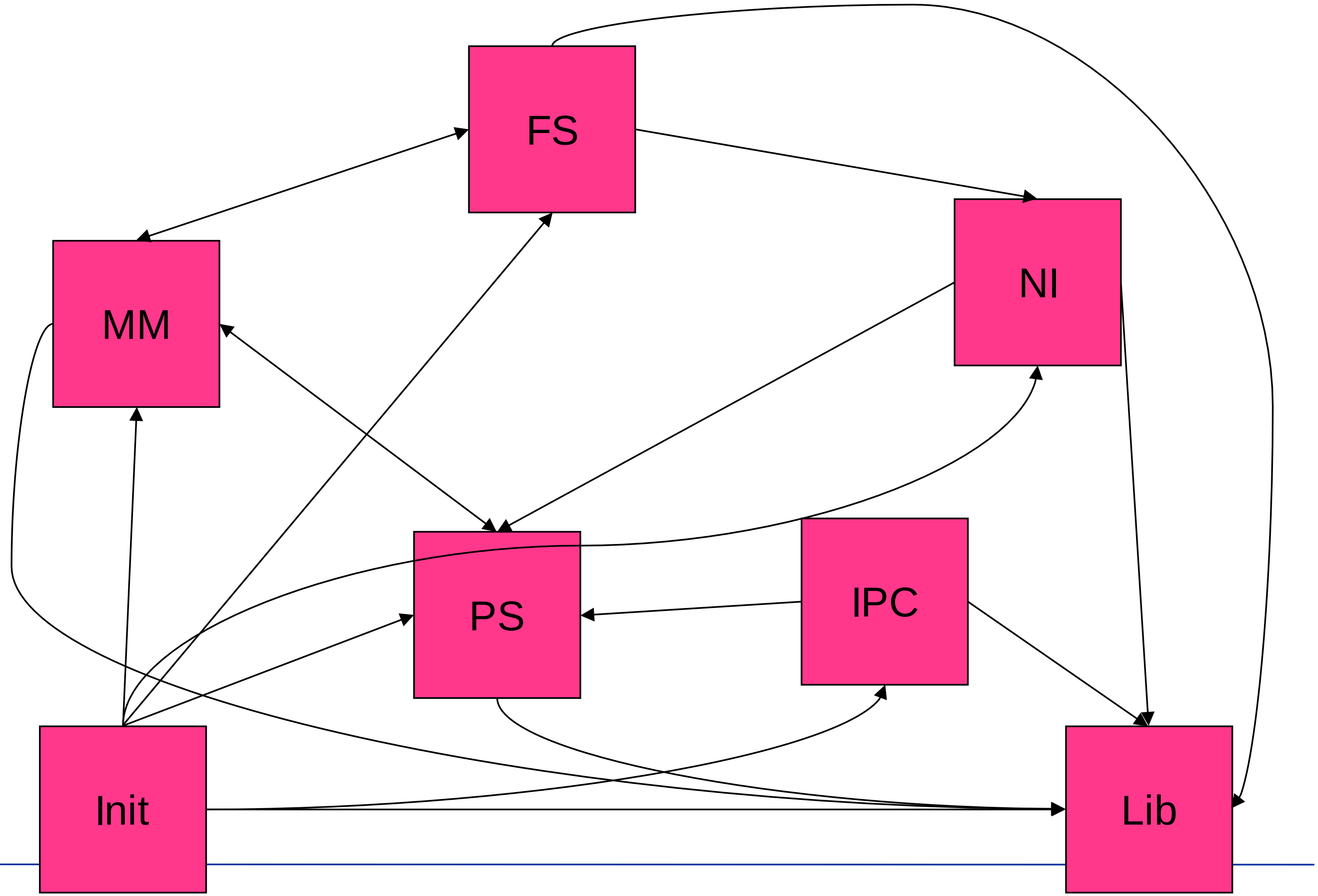
Logical View



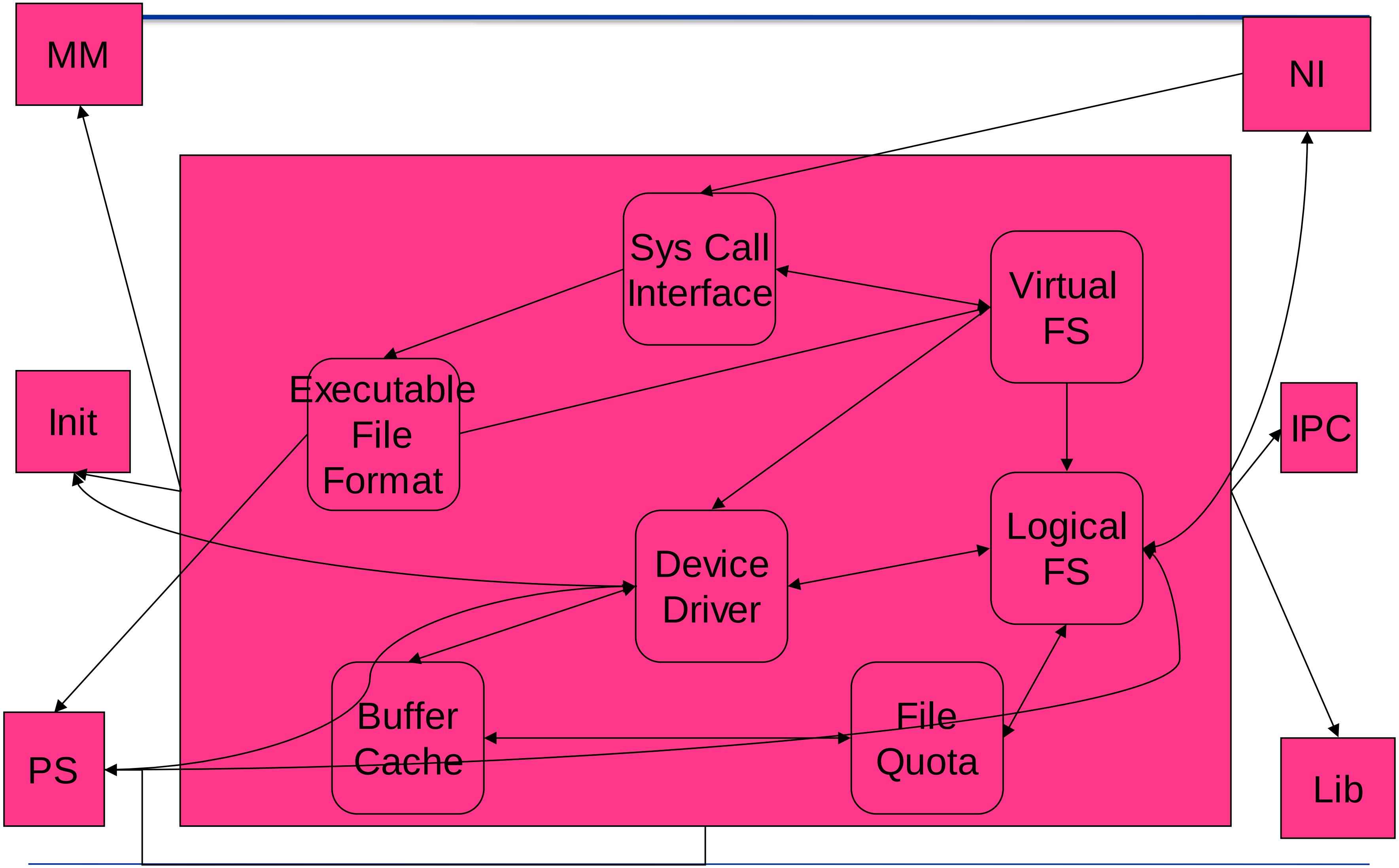
File System Broken Down –Logical View



Process View

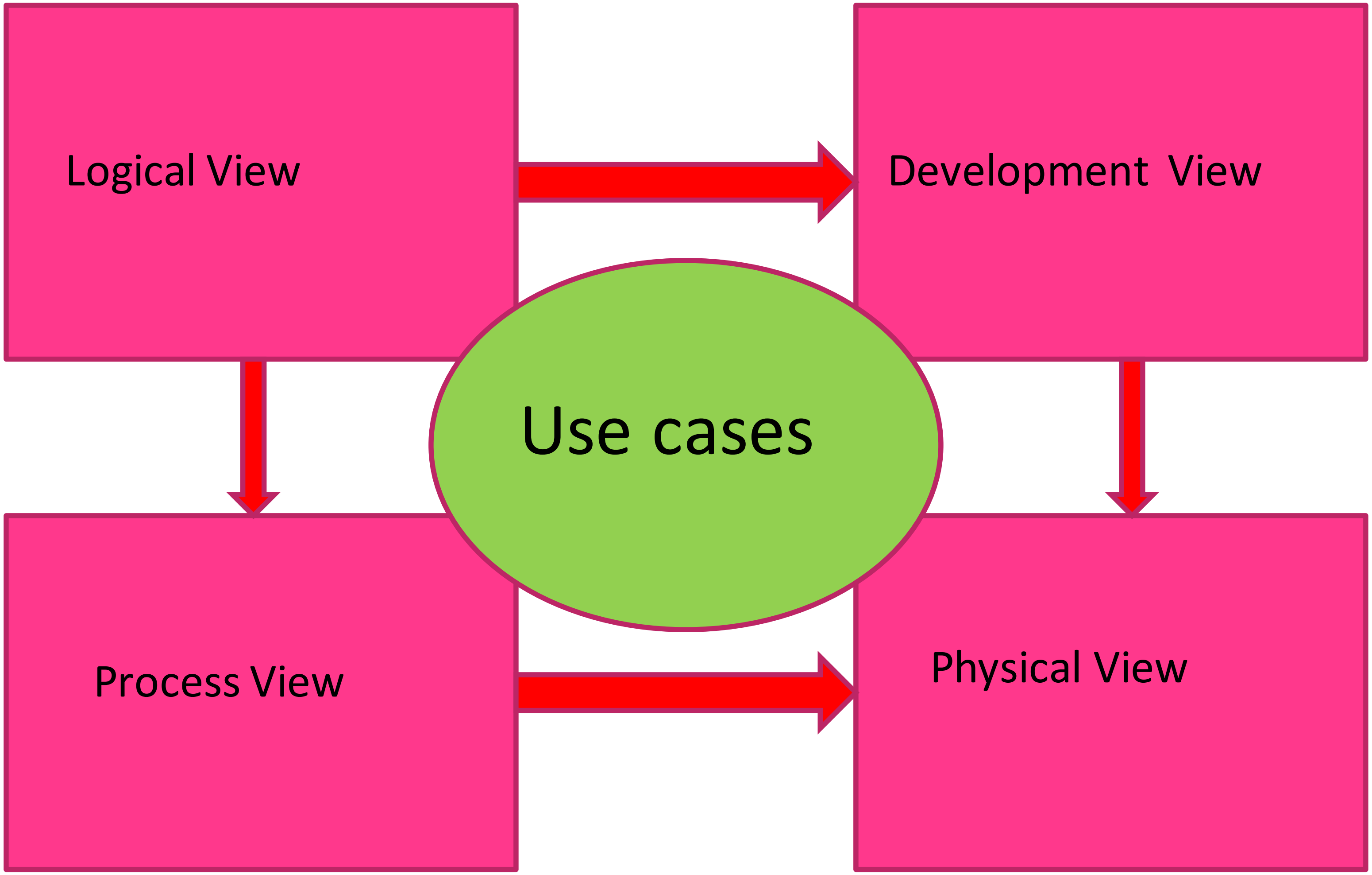


Breakdown of FS – Process View



4 + 1 view

- Philippe Kruchten, Rational Software, *Architectural Blueprints - The 4+1 View Model of Software Archtecture*, IEEE Software, 1995
  - Use case view
  - Logical view
  - Process view
  - Implementation view
  - Deployment view



Logical View

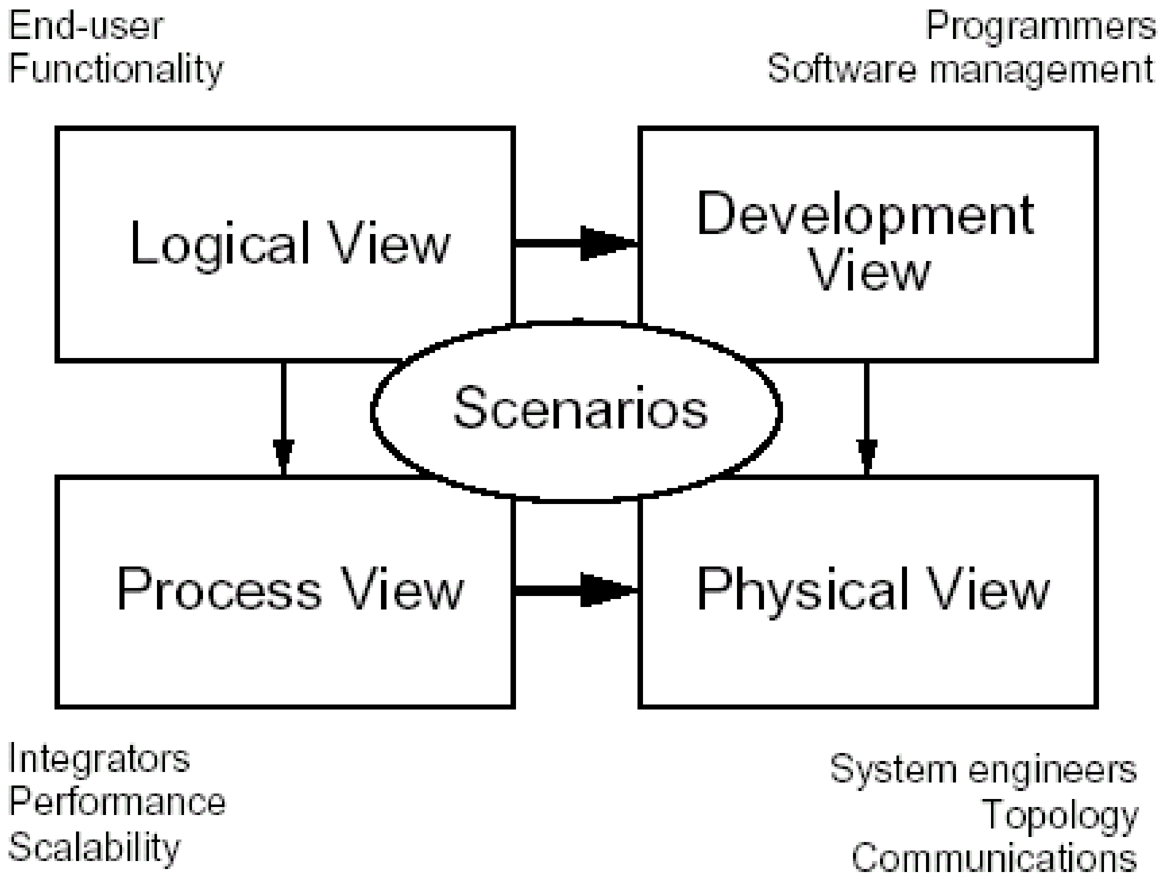
Development View

Use cases

Process View

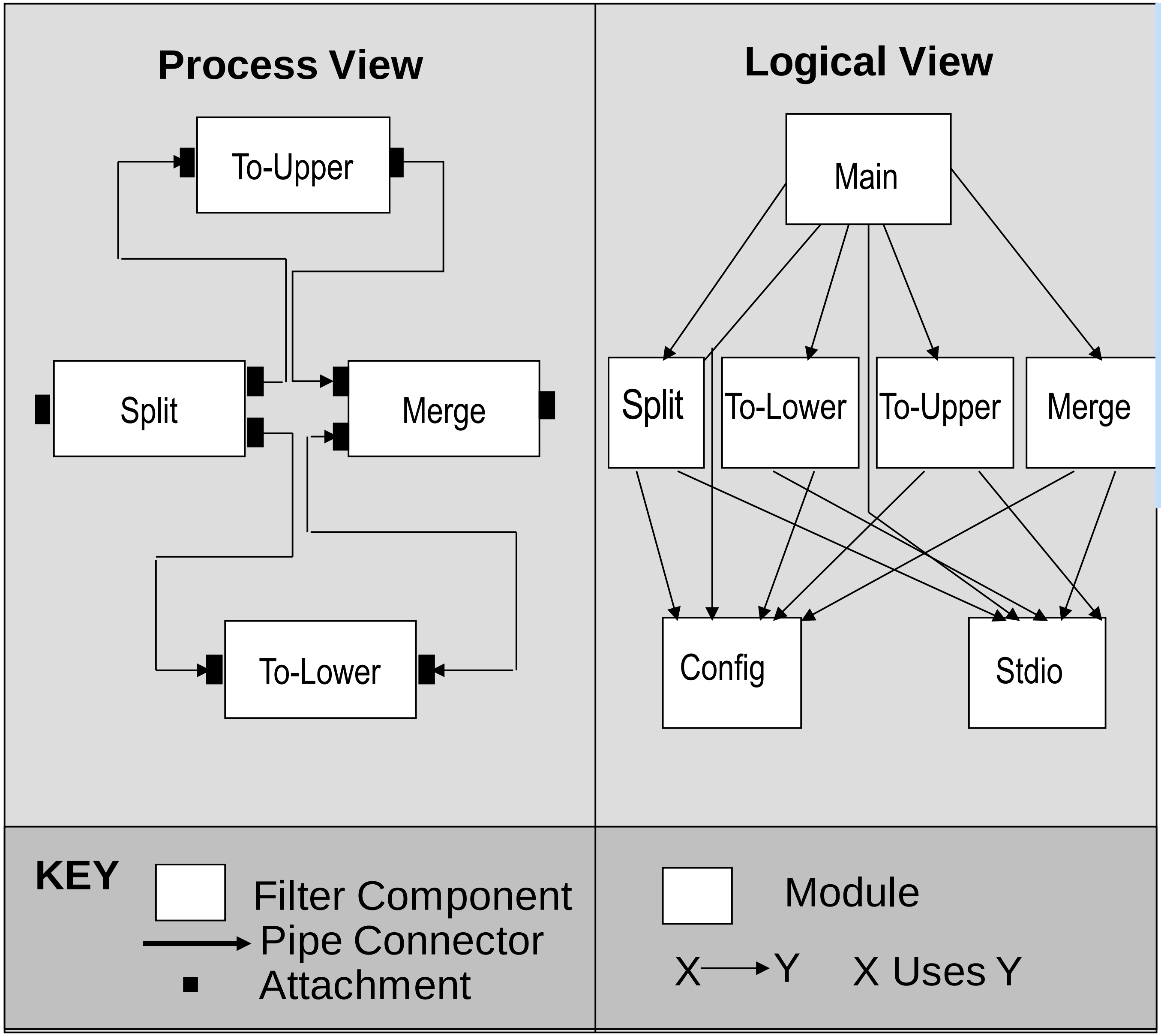
Physical View

4+1 view


End-user
Functionality
Programmers
Software management
Logical View
Development View
Scenarios
Process View
Physical View
Integrators
Performance
Scalability
System engineers
Topology
Communications

What do the views do?

- *logical* view talks about the decomposition of the system into its constituent modules
- *process* view captures the execution time components and how they interact
- *physical* view which describes the relationship between the software components and the hardware
- *development* view deals with development time issues like files, people
- illustrated by a few selected *use cases*, or *scenarios*



C&C and module views of the Capitalize System. The simple system illustrated here accepts as stream of characters as input and produces a new stream

of characters identical to the original but with upper-case and lowercase characters alternating

Home Work

Look at every diagram you see from this perspective