

Data Flow Diagram

Analisis Desain Sistem Informasi



Data Flow Diagram

The use of DFD as a modeling tool was popularized by DeMarco & Yourdon (1979) and Gane & Sarson (1979), using a Structured Systems Analysis methodology.

DFD illustrates the flow of data within an information system, whether an existing system or a new system, from a logical perspective without considering the physical environment in which the data resides.

DFD is a system design tool that focuses on data flow and is easy for programmers to understand and communicate.



Symbols Used

Process



Data flow



Data store



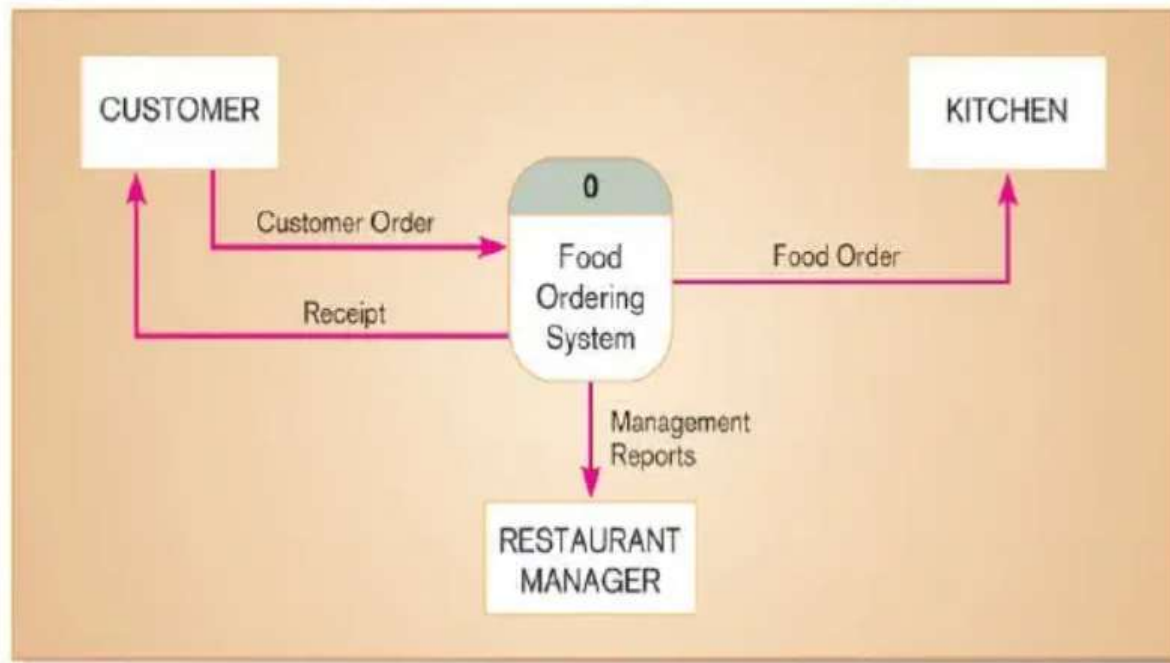
External
entity



Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdan Symbol
Every <i>process</i> has - A number - A name (verb phase) - A description - One or more output data flows - Usually one or more input data flows	Label (name) - Type (process) - Description (what is it) - Process number - Process description (Structured English) - Notes		
Every <i>data flow</i> has - A name (a noun) - A description - One or more connections to a process	Label (name) - Type (flow) - Description - Alias (another name) - Composition (description of data elements) - Notes		
Every <i>data store</i> has - A number - A name (a noun) - A description - One or more input data flows - Usually one or more output data flows	Label (name) - Type (store) - Description - Alias (another name) - Composition (description of data elements) - Notes		
Every <i>external entity</i> has - A name (a noun) - A description	Label (name) - Type (entity) - Description - Alias (another name) - Entity description - Notes		

Context Diagram

Figure 7-4 Context diagram of Hoosier Burger's food ordering system

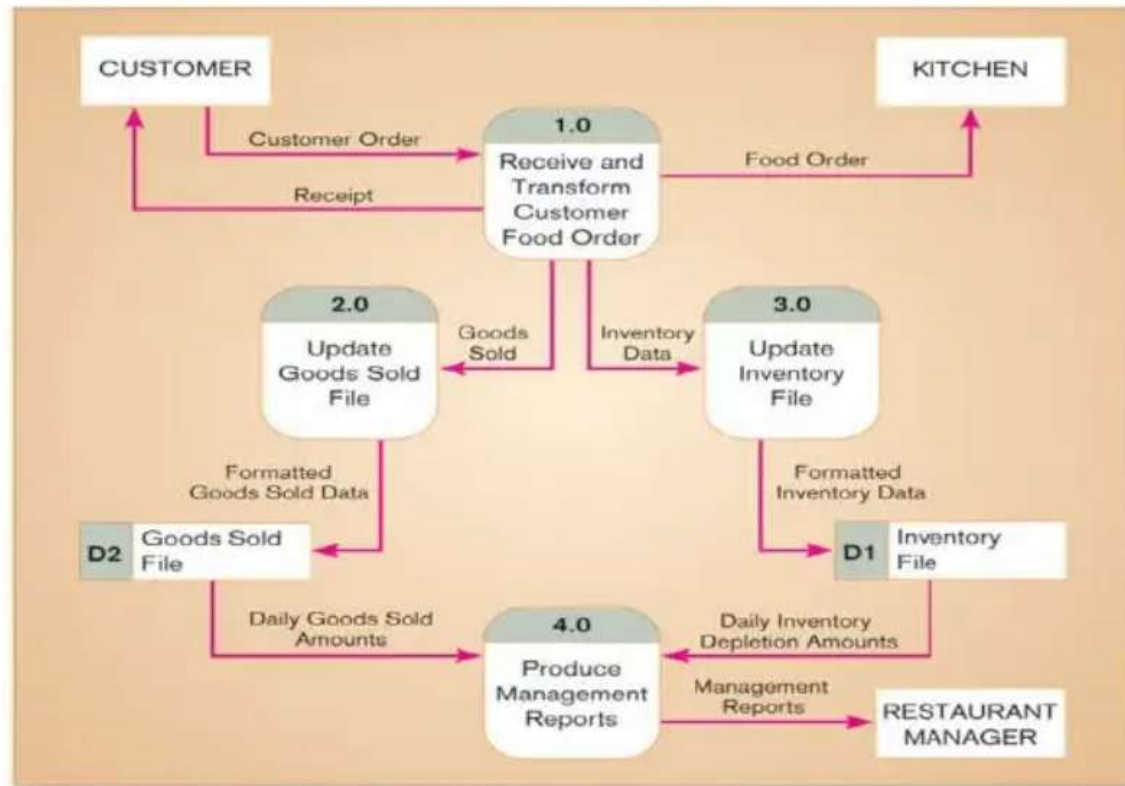


Context diagram shows the system boundaries, external entities that interact with the system, and major information flows between entities and the system.

NOTE: only one process symbol, and no data stores shown.

DFD Level 0

Figure 7-5 Level-0 DFD of Hoosier Burger's food ordering system

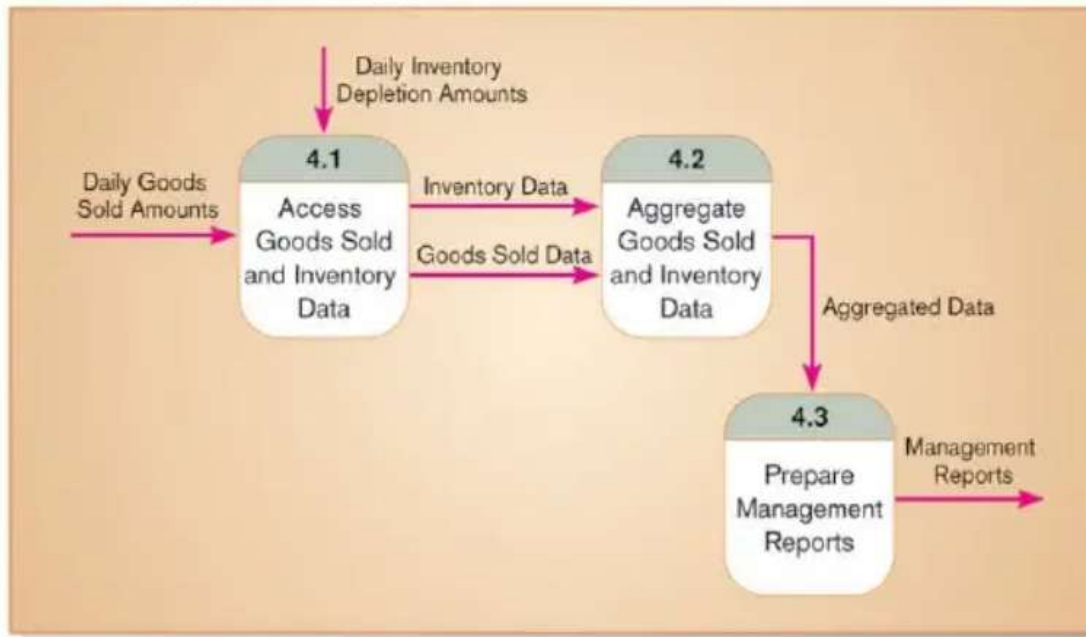


Level-0 DFD shows the system's major processes, data flows, and data stores at a high level of abstraction.

Processes are labeled 1.0, 2.0, etc. These will be decomposed into more primitive (lower-level) DFDs.

DFD Level 1

Figure 7-8 Level-1 diagram showing the decomposition of Process 4.0 from the level-0 diagram for Hoosier Burger's food ordering system



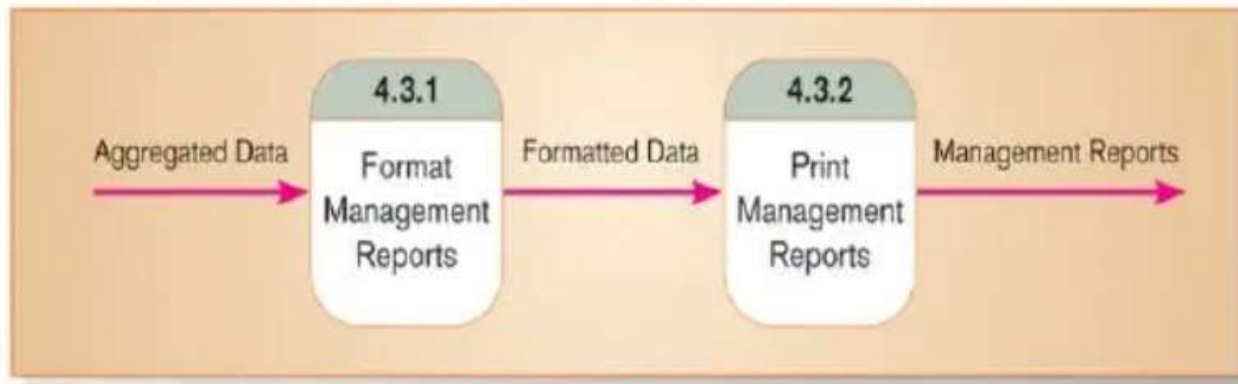
Level-1 DFD shows the sub-processes of one of the processes in the Level-0 DFD.

This is a Level-1 DFD for Process 4.0.

Processes are labeled 4.1, 4.2, etc. These can be further decomposed in more primitive (lower-level) DFDs if necessary.

DFD Level-n

Figure 7-9 Level-2 diagram showing the decomposition of Process 4.3 from the level-1 diagram for Process 4.0 for Hoosier Burger's food ordering system



Level- n DFD shows the sub-processes of one of the processes in the Level $n-1$ DFD.

This is a Level-2 DFD for Process 4.3.

Processes are labeled 4.3.1, 4.3.2, etc. If this is the lowest level of the hierarchy, it is called a *primitive DFD*.

Requirements for Creating a DFD



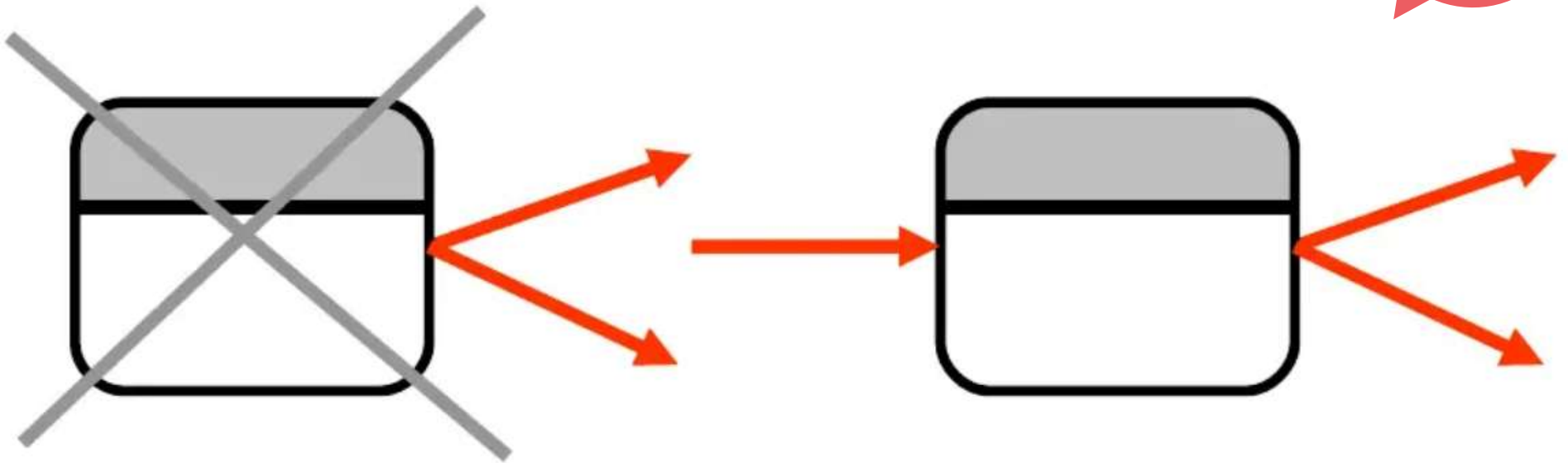
A properly designed DFD should be visually appealing, easy to read, and easy for users to understand.

The requirements for creating a DFD are as follows:

1. Use appropriate names for each DFD component.
2. Number the process components appropriately.
3. Arrange the DFD clearly and in a visually pleasing manner.
4. Avoid creating overly complex DFD diagrams.
5. Ensure that the DFD accurately represents the system being developed.

DFD Rules

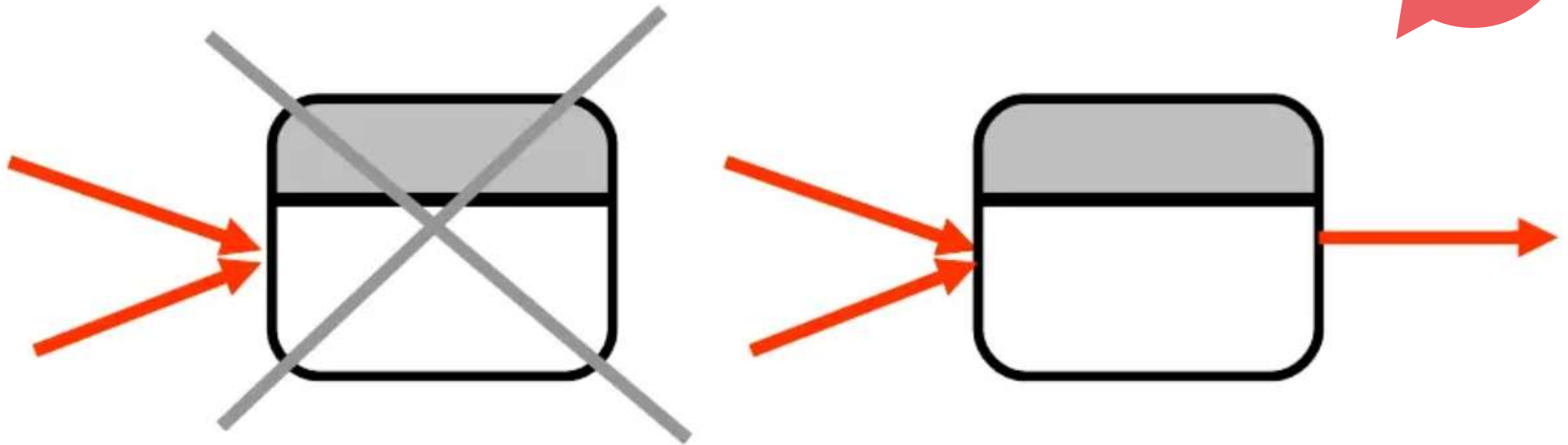
1



No process can have only outputs

DFD Rules

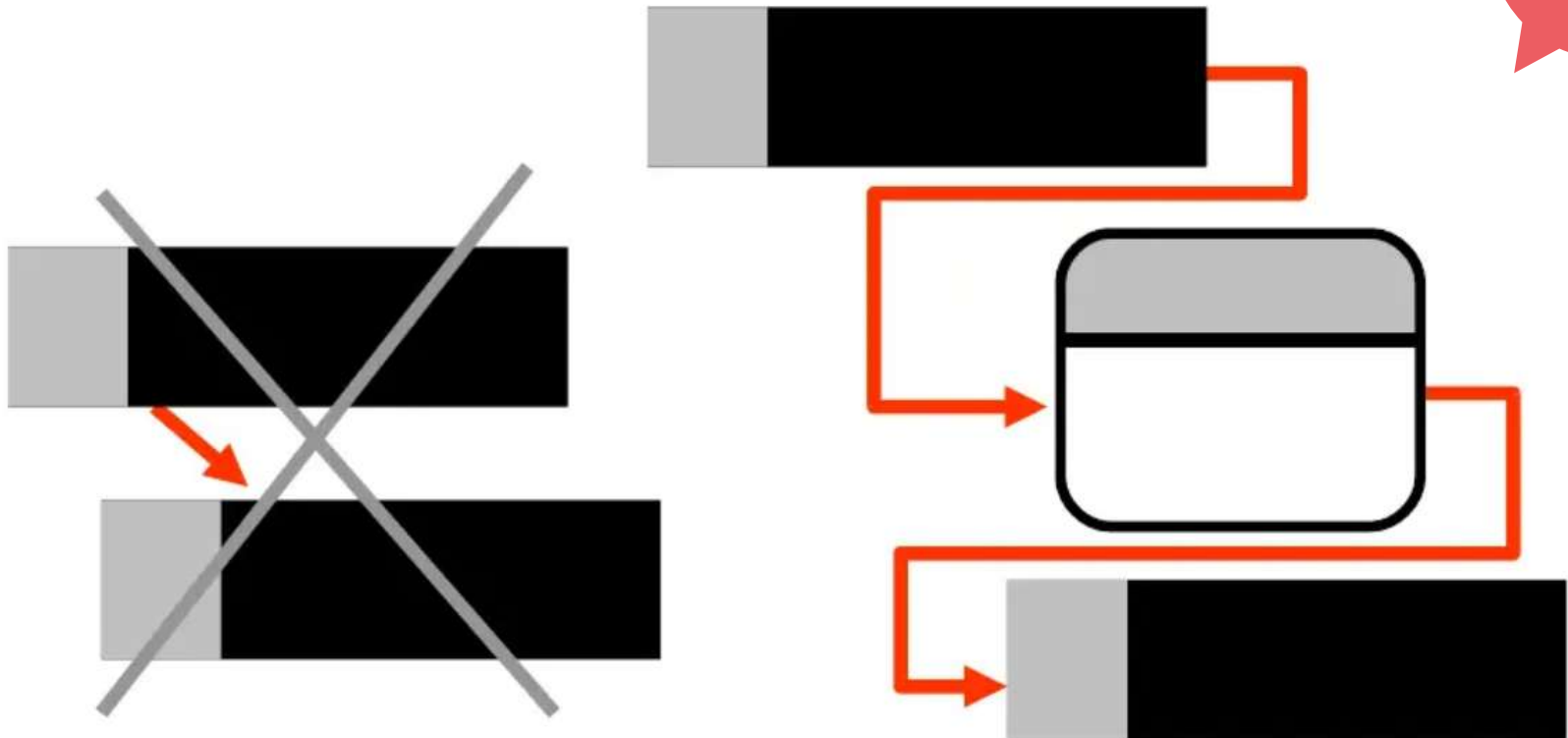
2



No process can have only inputs

DFD Rules

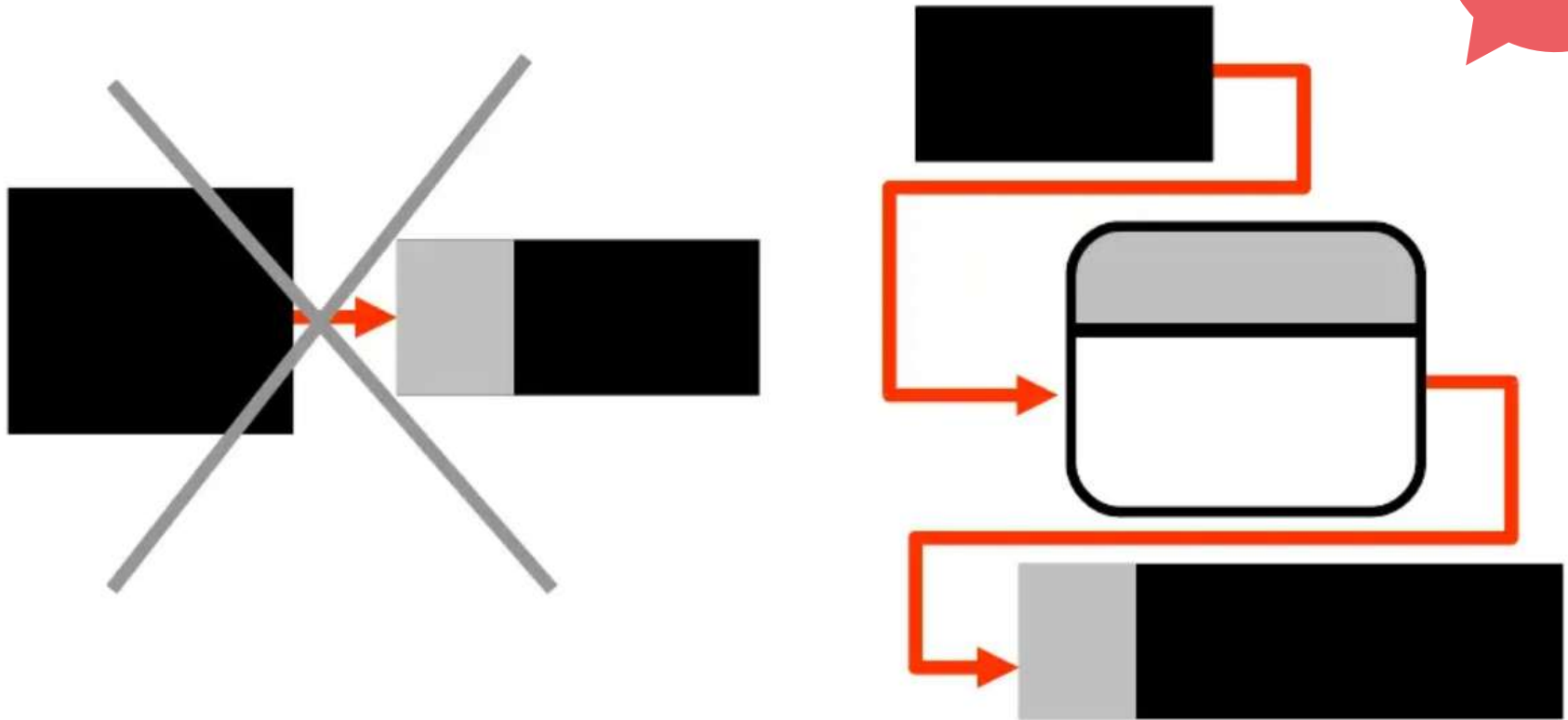
3



Data cannot move directly from one data store to another

DFD Rules

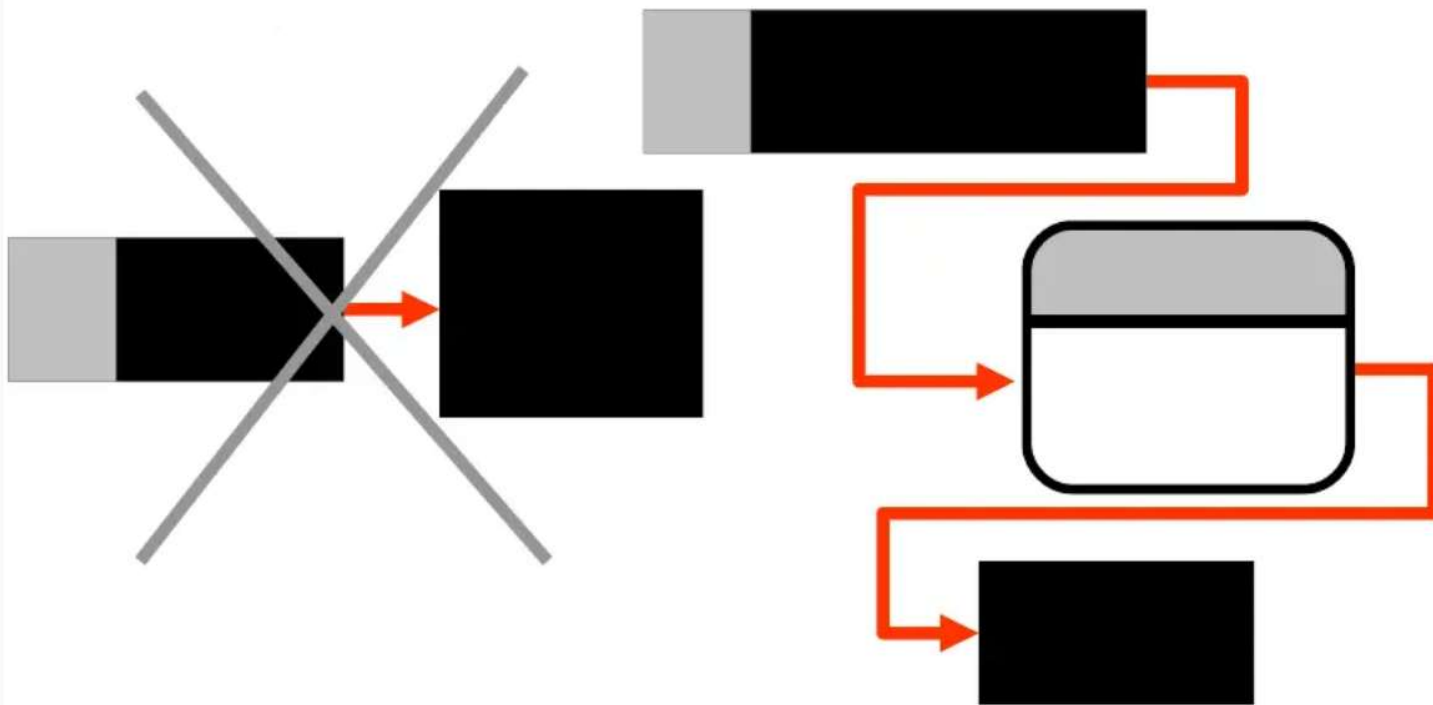
4



Data cannot move directly from an external entity to a data store

DFD Rules

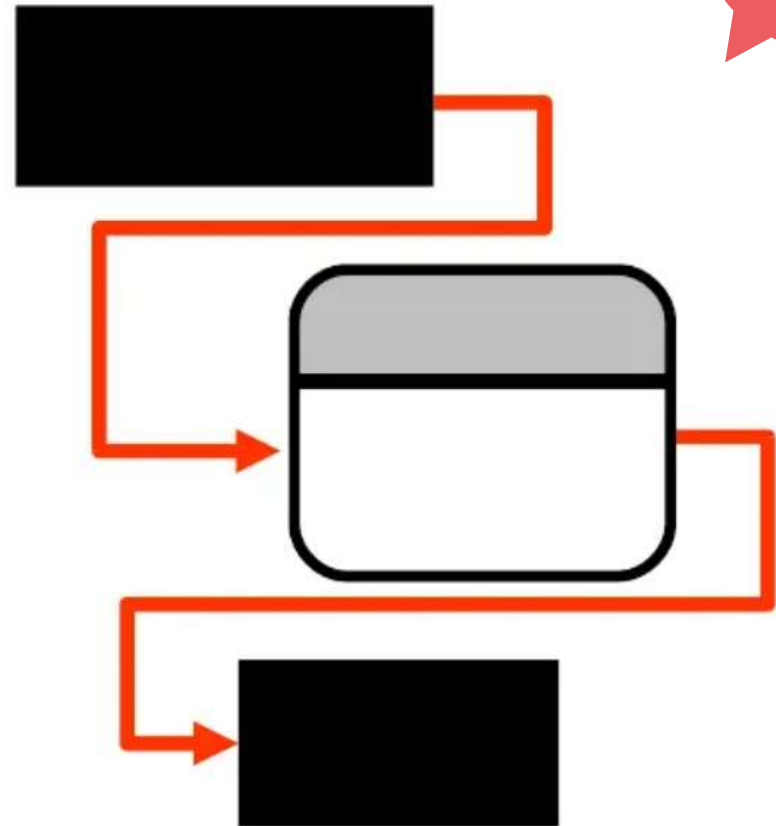
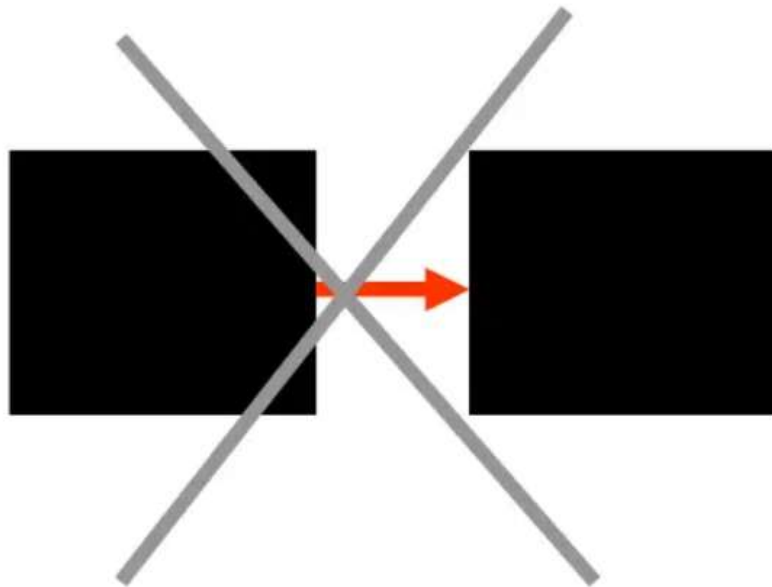
5



Data cannot move directly from a data store to an external entity (sink)

DFD Rules

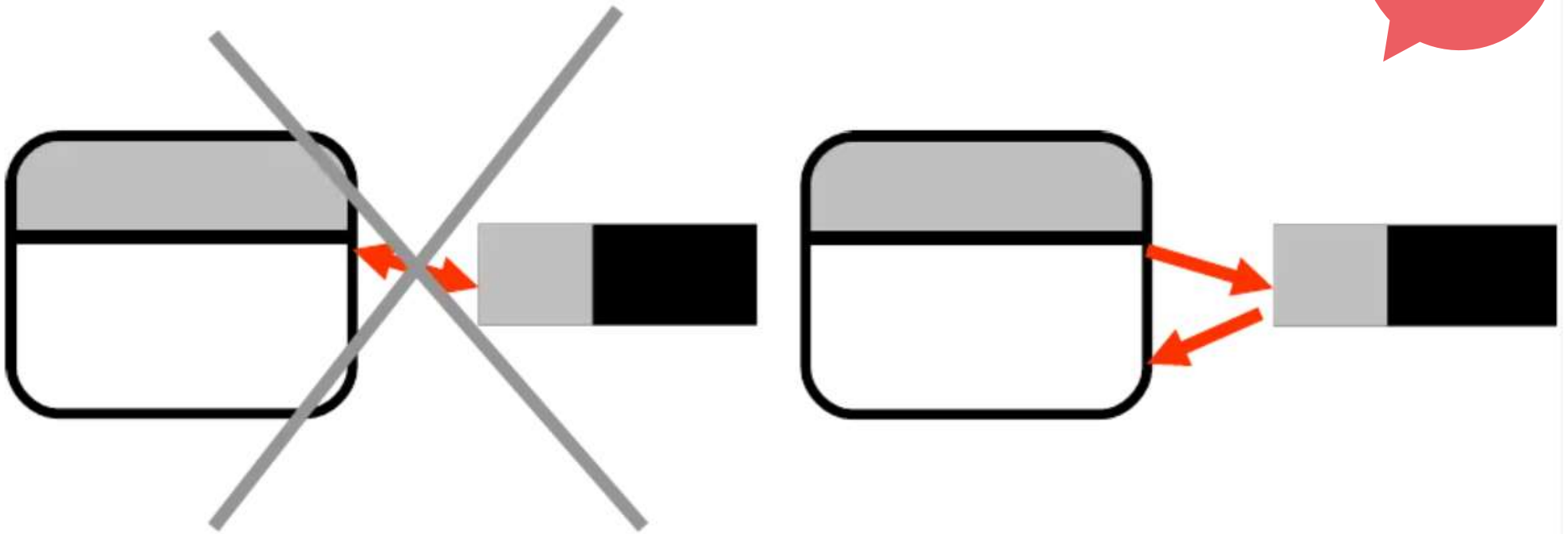
6



Data cannot move directly from an external source to an external sink

DFD Rules

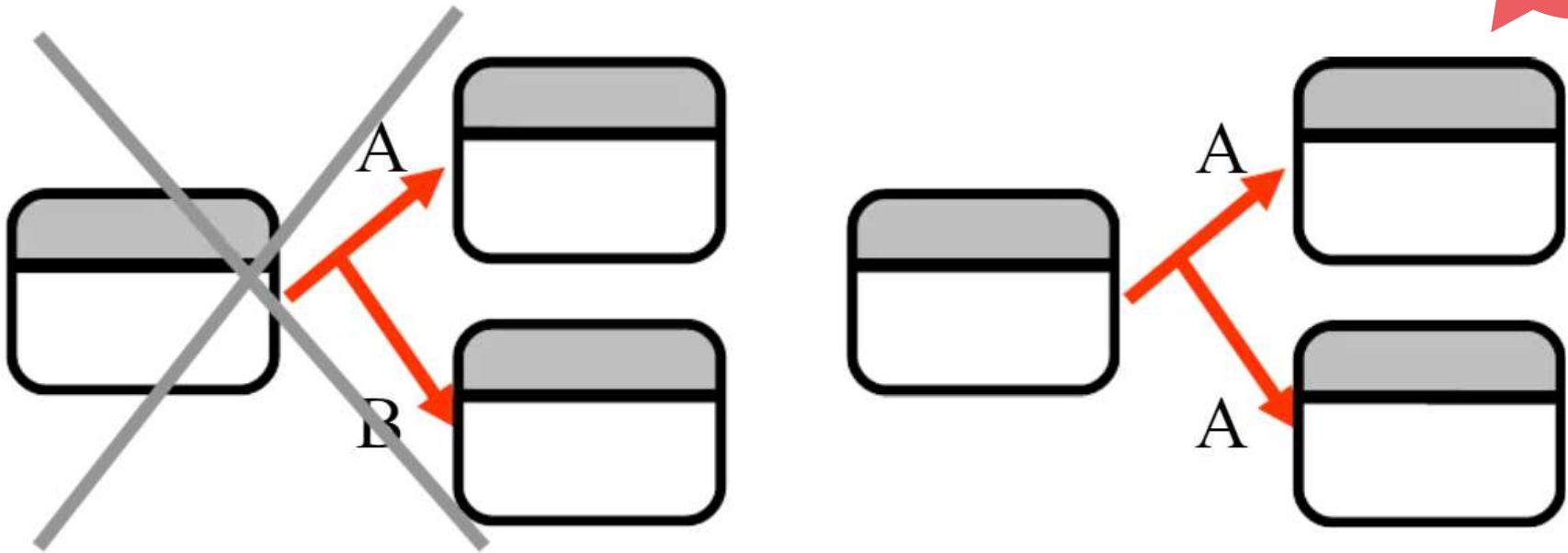
7



A data flow has only one direction of flow between symbols

DFD Rules

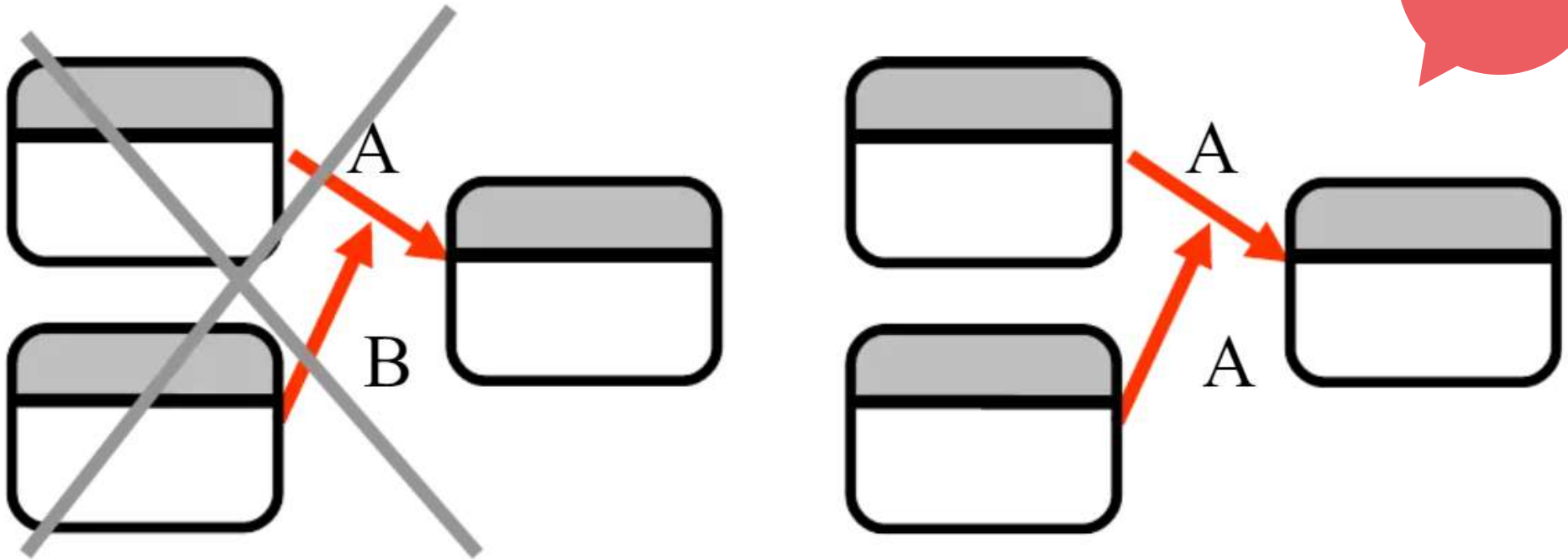
8



A fork in data flow means only the exactly same data goes from a common location to two or more different processes

DFD Rules

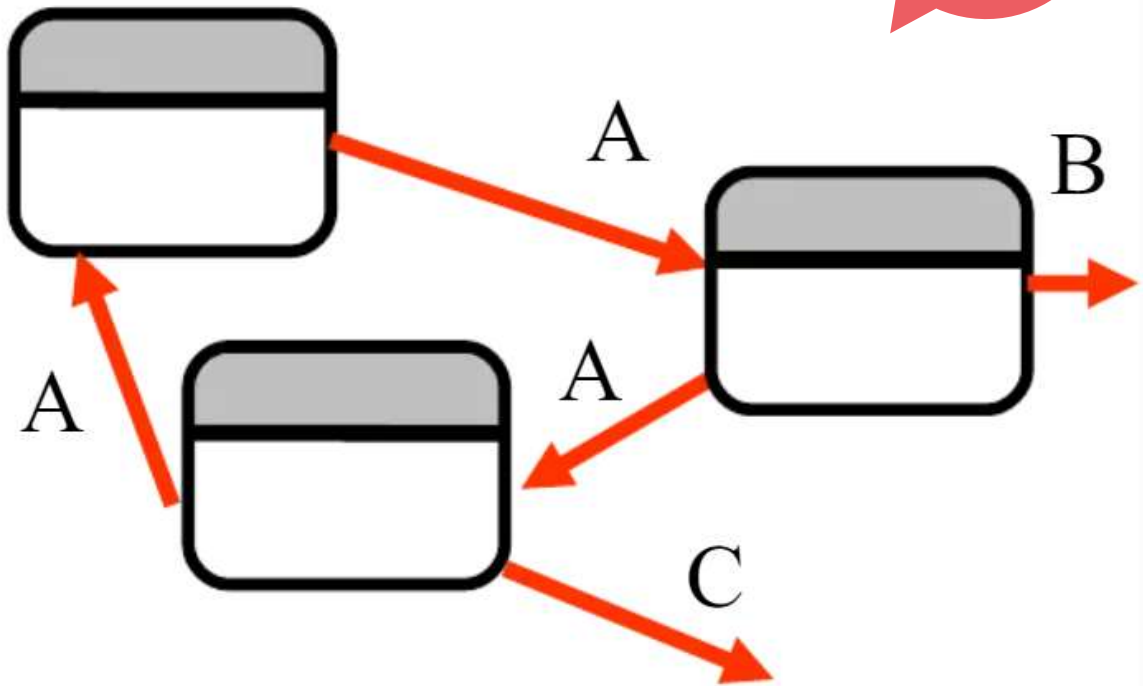
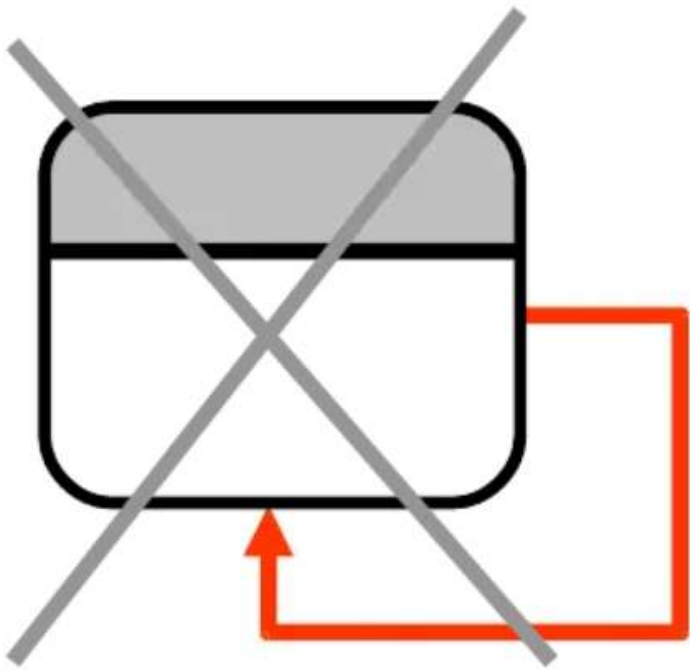
9



A join in a data flow means that exactly same data goes from any of two or more different processes, data stores to a common location

DFD Rules

10



No recursive data flow on a process

DFD Rules

11



A data flow to a data store means update (delete or change)

DFD Rules

12



A data flow from a data store means retrieve or use

THANK YOU
^^

