Traditional Switch Process: a traditional switch would contain

- Data Plane: handles incoming traffic and forwards it based on decisions by control plane
- Control Plane: makes decisions about where and how traffic is forwarded

(Modern) Software-Defining Networking: (SDN): (modern way)

- Idea of decoupling the data and control planes in a switch and moving the control plane only to general purpose hardware
  - Software-based control runs on general purpose hardware : centralises the control plane
- Therefore, switches will only have hardware for data plane (to forward)
- Advantages: cost of individual switches will be cheaper

(Modern) Software Defining Network Architecture Layers/Planes:

- Infrastructure Layer (bottom): includes switches - controls data plane
  - *South bound application programming interface (API)*: Connects individual networking devices to their software based controllers (think above control panel)
- Control Layer: includes controllers itself – manages policies and network traffic
- Application layer:
  - Communicates with SDN controllers in the control layer via *northbound application programming interface (API)*

Overflow: (think first version of sdn)

- Early iterations of SDN from 2008-2009 managed by open network foundation (open standard) for southbound API
- Really a link layer protocol specifying:
  - Separation between control and data plane
  - Interface between switches and network os (controller)
  - Use existing structures such as routing and forwarding tables in hardware

Openflow operation:

- Each openflow device in network maintains flow tables
- Flow tables: contains list of rules that dictate how packets are handled specially:
  - Entries for hosts (think mac address table for switches)
  - Actions to be applied to matching traffic (think similar mac address identified)

- When openflow switch receives a frame they will try to match against flow tables based on **rules** in the different tables (think frame received match with 30 tables all with different answers to rules)
  - Rules: specify header fields to match (ie ip or destination mac or port etc) or receiving interface
- If match found then specified action for corresponding flow table and corresponding rule will be performed

- If no match for packet on flow tables entries, switch will perform **table-miss action**
    - Table-miss action: will usually specify to switch forward non-matching packets to controller or drop the packets

Disadvantage of a centralised control plane: (in openflow + modern)

- A single centralised control plane/SDN controller is bad since if it goes down then-
    - For openflow design the whole switching process would cease
    - Network may be able to continue to operate under previous known state but no new rules can be specified
- Heavy reliance on SDN controller **adds latency** to traffic: switches must wait for controller to provide instructions (slower performance/process/delay)
- Therefore different approaches to reduce redundancy of single controllers

Approach: Hybrid approach: (not as cheap)

- Switches now have control panel back (some of functions) but controllers are still used and there
- Controllers specifies and propagates polices to switches
- Advantages
    - Failure means no changes can be made to network, but no consequences otherwise
    - Reduce latency by no longer being reliant on controller for forwarding decisions
- Examples of Hybrid approach- **cisco application centric infrastructure**