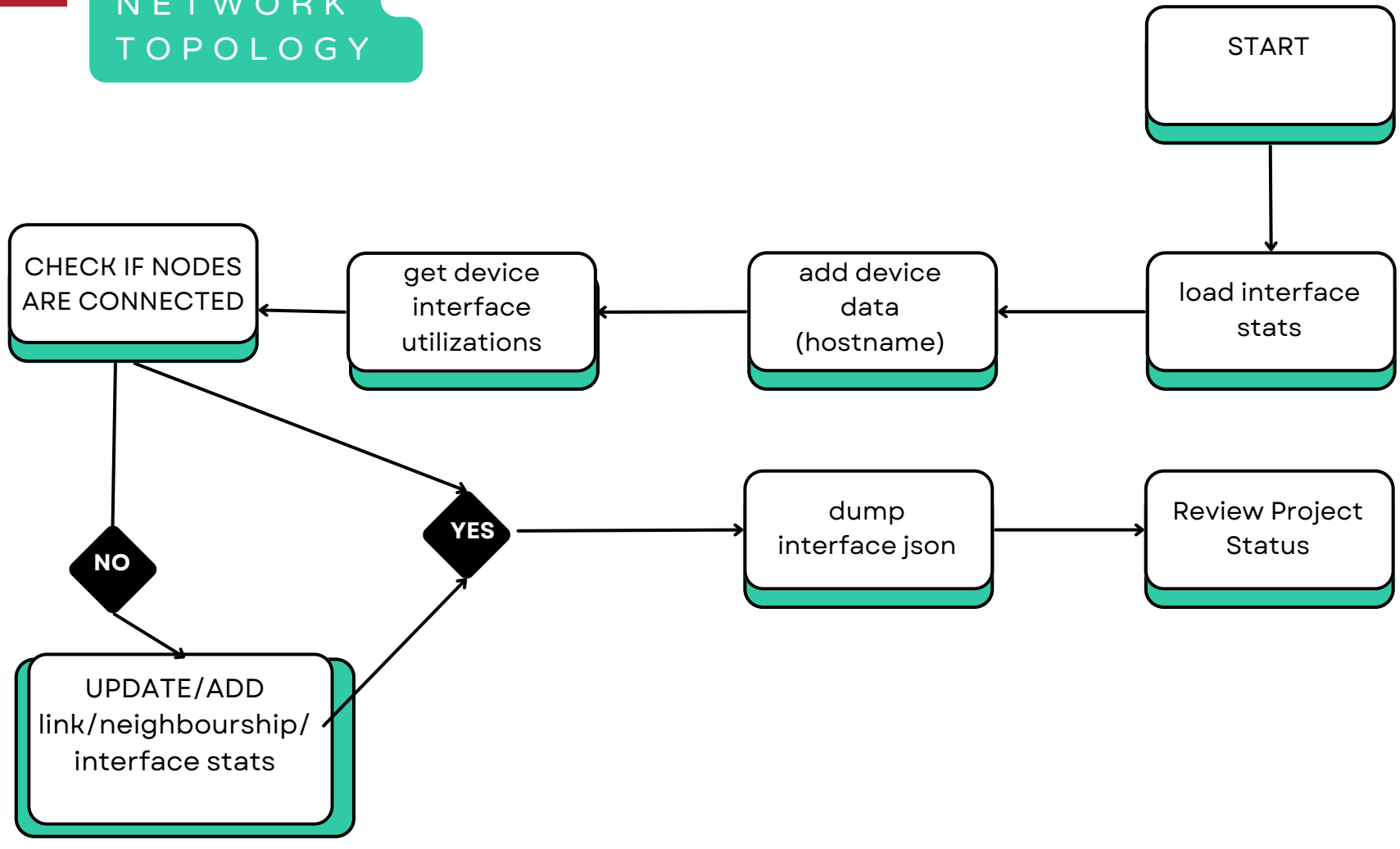
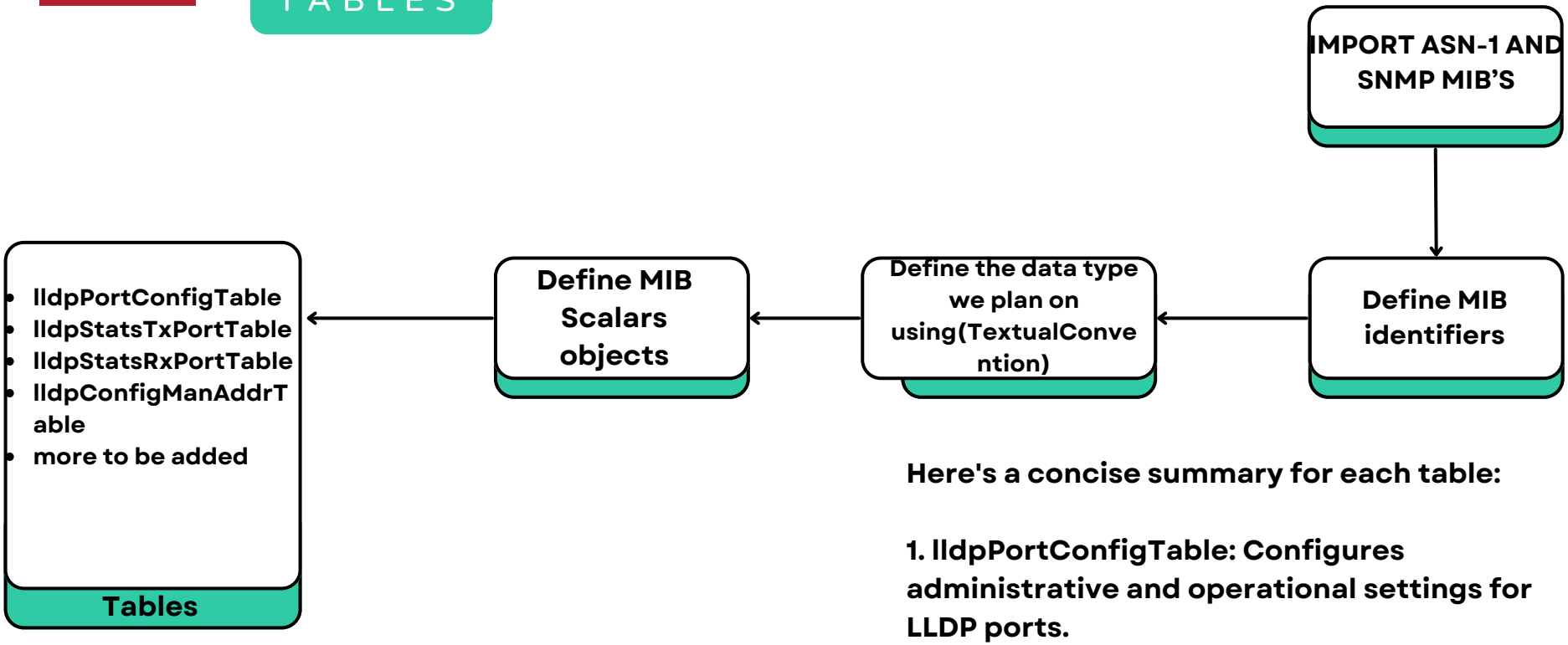
WELLS FARGO

# NETWORK TOPOLOGY



WELLS FARGO

# LLDP-MIB TABLES




**Here's a concise summary for each table:**

**1. lldpPortConfigTable: Configures administrative and operational settings for LLDP ports.**

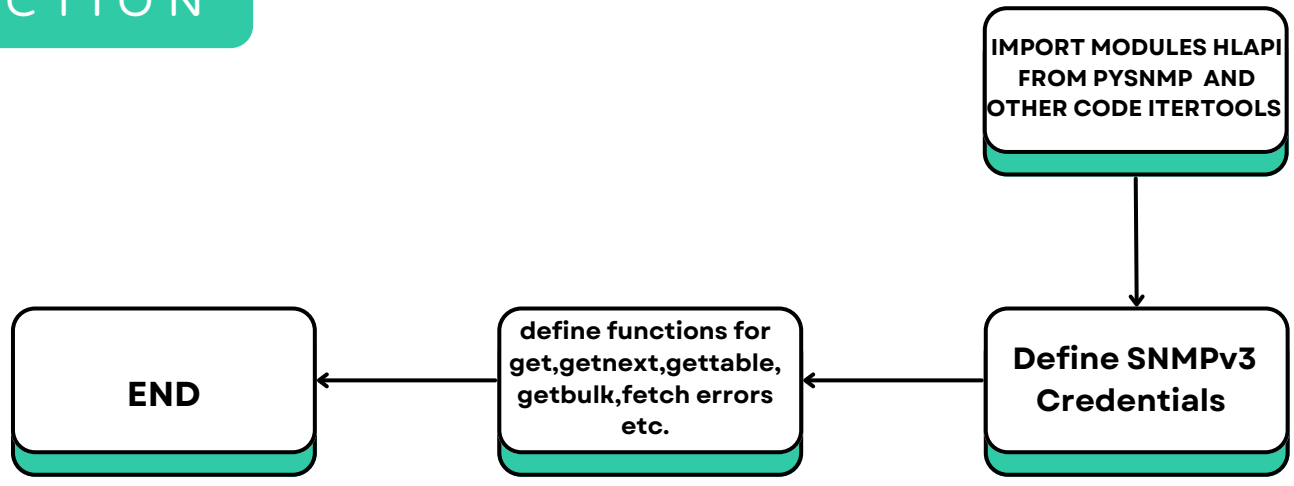
**2. lldpConfigManAddrTable: Specifies management addresses and their associated transmitting ports.**

**3. lldpStatsTxPortTable: Tracks statistics for LLDP frames transmitted by each port.**

**4. lldpStatsRxPortTable: Tracks statistics for LLDP frames received by each port.**

WELLS FARGO

# SNMP EXTRACTION



WELLS FARGO

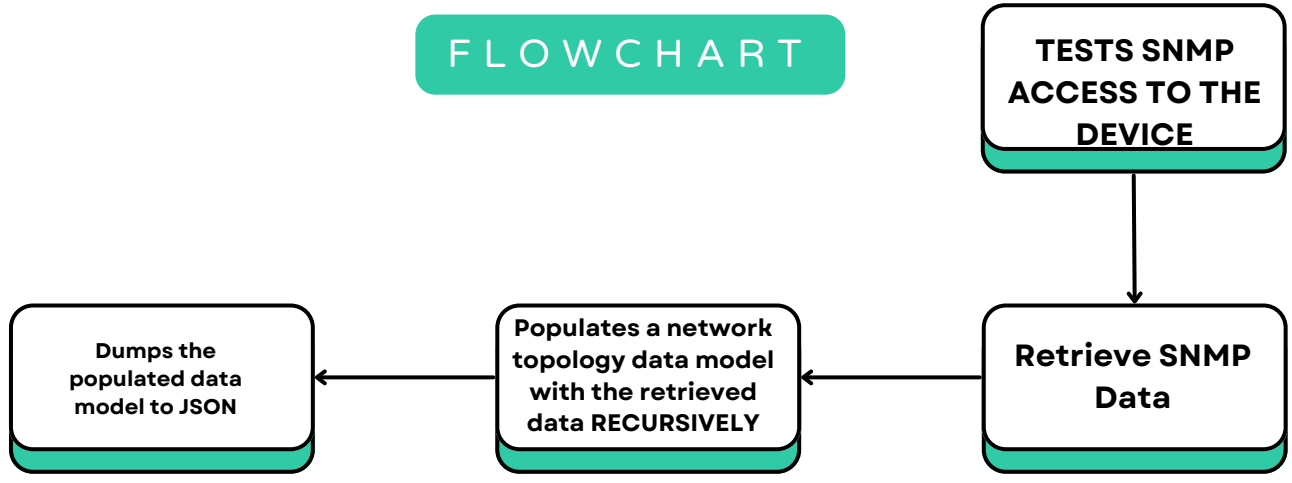
# PARSING THROUGH THE DATA

1. PARSE ARGUMENTS : USE ARGPARSE TO HANDLE COMMAND-LINE ARGUMENTS, INCLUDING --VERSION AND --REPEAT.
2. CHECK REPEAT ARGUMENT:
3. IF REPEAT IS 0, CALL THE MAIN_WITH_ARGS() FUNCTION ONCE.
4. IF REPEAT IS GREATER THAN 0, ENTER A LOOP TO CALL MAIN_WITH_ARGS() AND THEN SLEEP FOR THE SPECIFIED NUMBER OF SECONDS BEFORE REPEATING.
5. INITIALIZE A NETWORKTOPOLOGY MODEL INSTANCE.
6. READ AND PARSE THE CONFIGURATION FILE WHICH WE WILL DEFINE
7. IF REQUIRED SECTIONS (DEVICES AND DEFAULT) ARE MISSING, LOG AN ERROR AND EXIT.
8. IF SECTIONS ARE PRESENT, PROCEED TO ITERATE OVER THE LISTED DEVICES.
9. FOR EACH DEVICE, LOG THE START OF ACCESS TESTING.
10. ATTEMPT TO GET THE DEVICE'S HOSTNAME USING SNMP.
11. ON SUCCESS, LOG THE DATA.
12. ON ERROR, LOG THE ERROR AND CONTINUE TO THE NEXT DEVICE.
13. IF SNMP ACCESS IS SUCCESSFUL, LOG THE CONNECTION SUCCESS AND PROCEED TO:
14. RETRIEVE THE INTERFACES TABLE USING SNMP.
15. ON SUCCESS, LOG THE INTERFACES DATA.
16. ON ERROR, LOG THE ERROR AND CONTINUE.
17. RETRIEVE THE LLDP TABLE USING SNMP.
18. ON SUCCESS, LOG THE LLDP DATA.
19. ON ERROR, LOG THE ERROR AND CONTINUE.
20. ADD THE RETRIEVED INTERFACES AND LLDP DATA TO THE NETWORKTOPOLOGY MODEL.
21. ONCE ALL DEVICES ARE PROCESSED, DUMP THE POPULATED TOPOLOGY MODEL TO JSON AND DELETE THE MODEL INSTANCE.
22. PRINT RETURN CODE: PRINT THE RETURN CODE AND EXIT THE SCRIPT. IF REPEAT WAS SPECIFIED, THIS PROCESS LOOPS.

WELLS FARGO

# JSON

FLOWCHART




- We will be using a queue to add node data and will be popping out edges recursively for the creation of the whole topology

WELLS FARGO

After getting the data of the network:

First the device will look if the destination is in the same network or not (by making arp broadcast request or check if destination is already present in its device arp table)

If not present in the same network:

Packet forwarding from device to its gateway :

The device first needs to know the mac address of the gateway router to form layer2 header and forward the packet so it will first send the arp request which will be taken by the switch, switch broadcasts this packet to each port and the gateway router will then unicast the arp response switch will learn this and store its mac address in mac table and will be forwarded to device. Device then creates layer2 header and then packet will be forwarded to gateway router.

Packet forwarding Between two layer3 devices (Router-Router) :

We will see the routing table of the router we are currently right now on. After that we will do longest prefix match with the available networks and find the next hop/interface that packet should follow accordingly. If the router is multipath configured, then in that case there might be more than one next hop here we will choose all the hops and proceed (helps in tracing all the paths) this recursively until we reach the destination. Also, we here must first find the mac address of the next hop to forward the packet so arp request will be sent to find the mac address and next hop router will respond with its mac address with the help of which we can create layer2 header and then packet will be forwarded to the next hop.