



SIMULATING NETWORK DEVICES WITH PYTHON

Matias Torchinsky (matt)
matts@gmx.co.uk

WHAT WE WILL SEE

- **Introduction**
- **Motivations and goals**
- **Coding and Live samples**
- **Conclusions**

“IN THE BEGINNING ...”

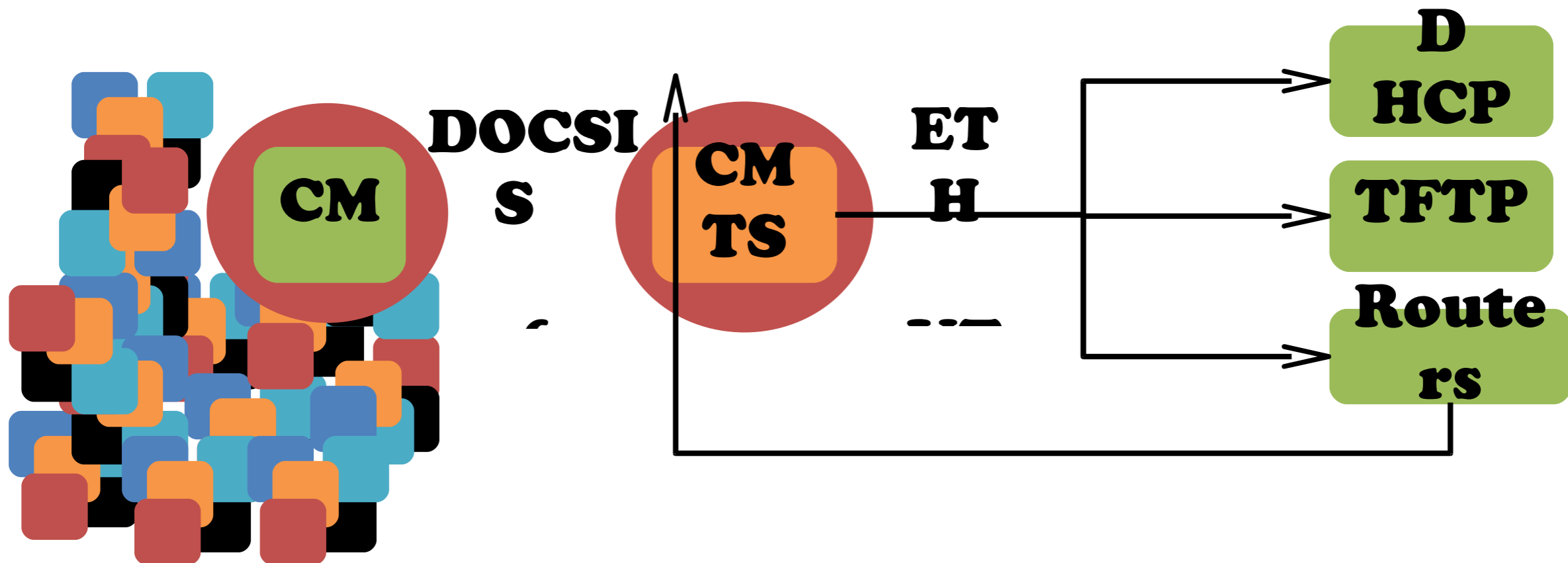
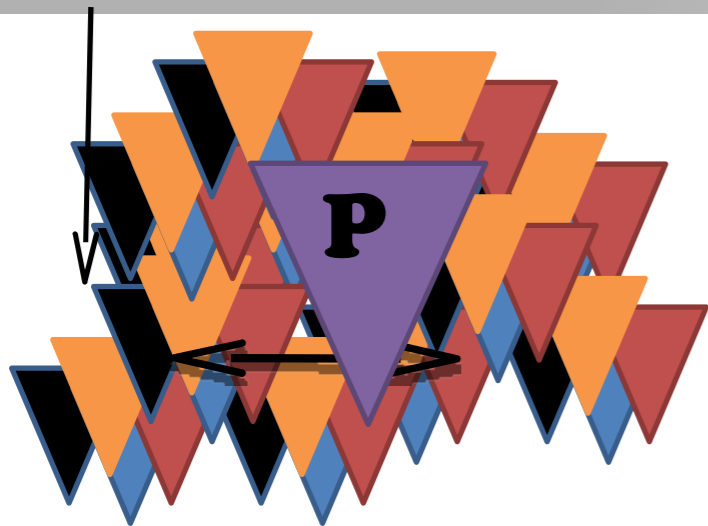


What is this ?

Why ?

What will we see ?



WELCOME TO THE ICD WORLD



SITUATION

/"PROBLEM TO SOLVE"/

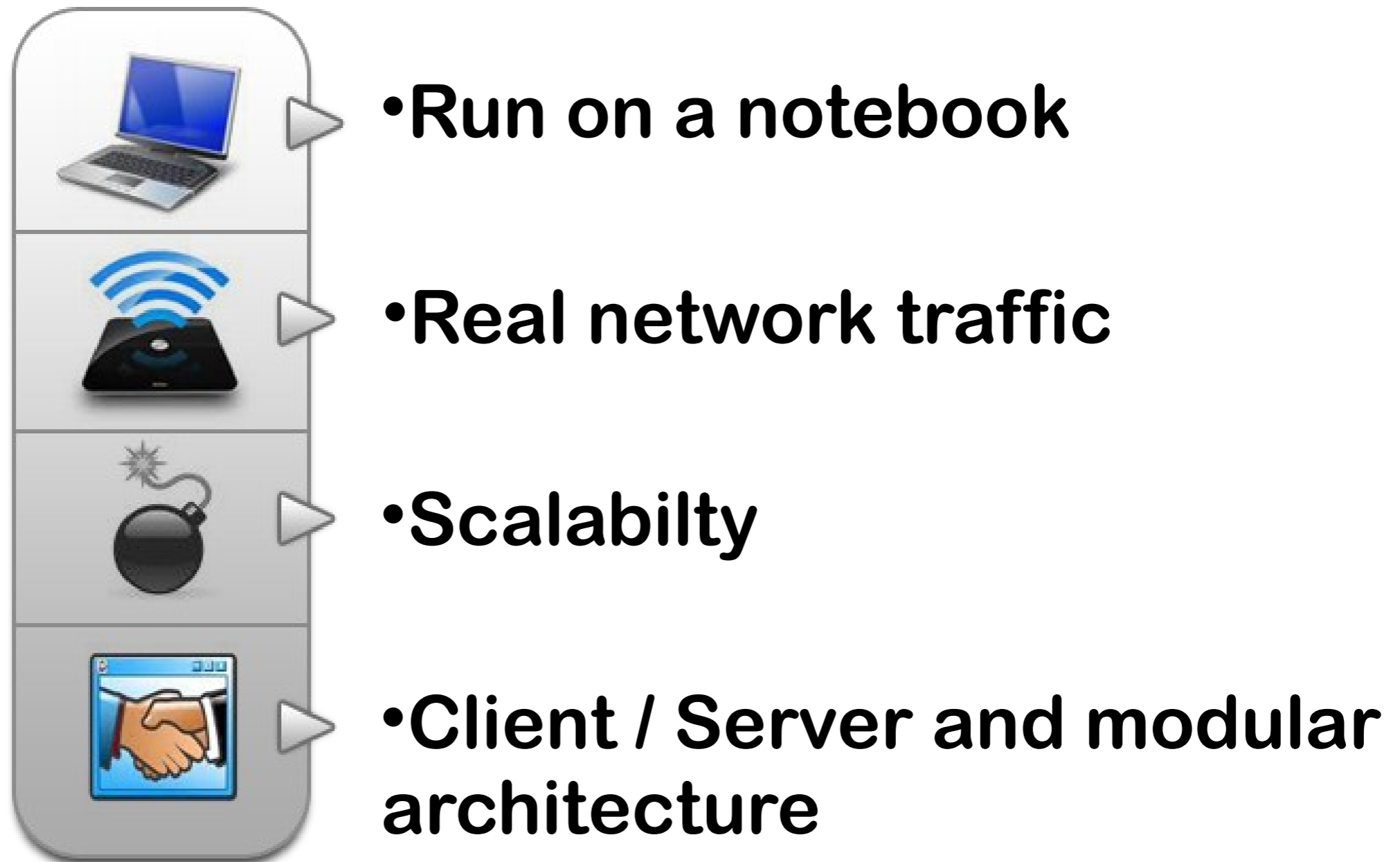


-  **No lab!**
- **Need critical mass!**
- **No open source simulator**
-  **Python or C++ ?**

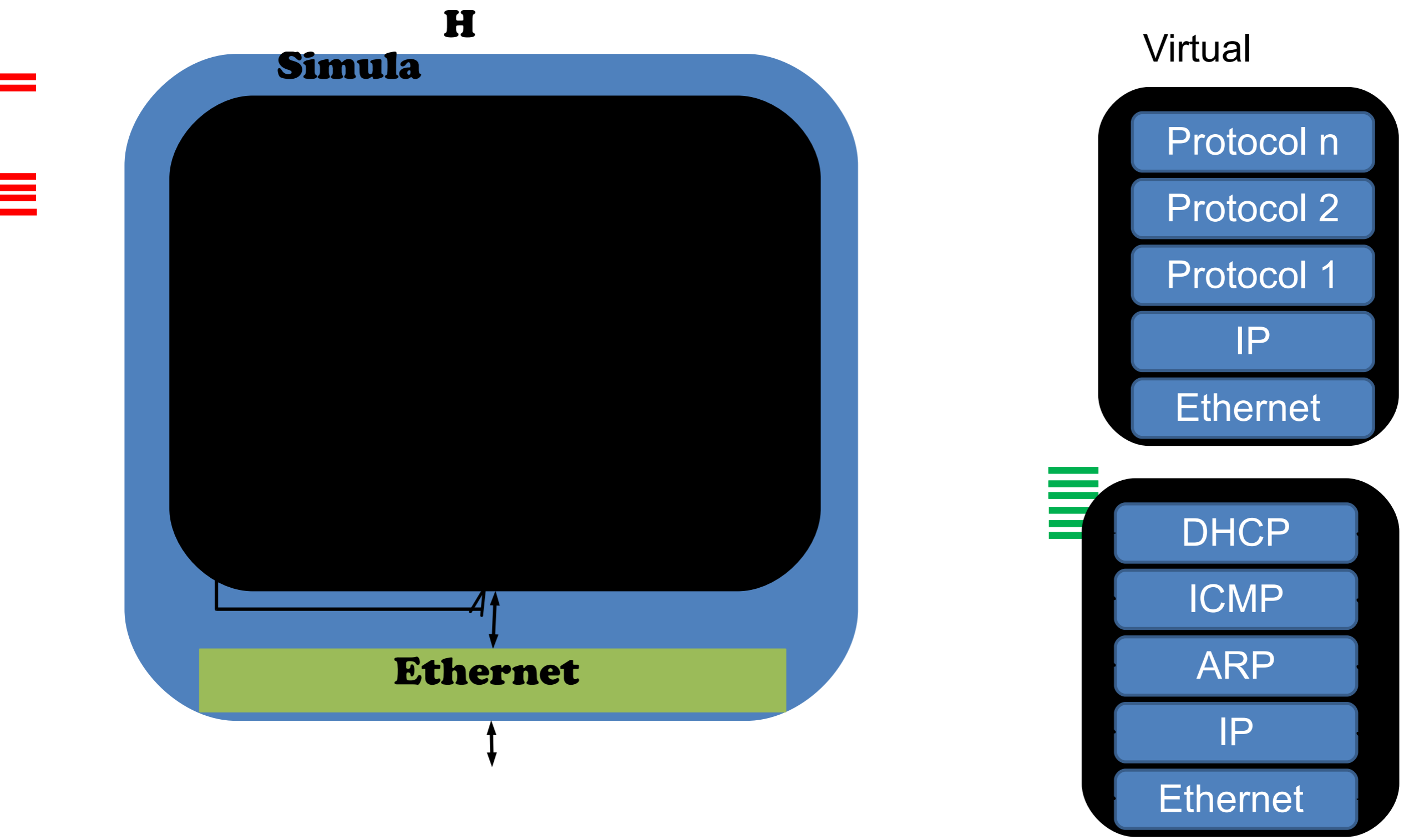
HANDS ON.... LET'S DEVELOP



SIMULATOR REQUIREMENTS



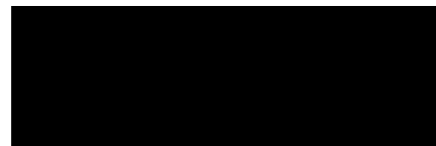
SIMULATOR ARCHITECTURE



CURRENT FEATURES - WHAT I HAVE DONE



- **Protocols supported:**



TFTP

ARP

ICMP

NTP

- **Link Layers :** **Ethernet**
- **Spoofing (Mac / IP)**
- **Event driven (threads scalability problem)**
- **Adding new protocols is quite easy ! (modular)**
- **API**

WHAT'S SO COOL ABOUT THIS PROJECT ?



- Lets see a live sample !
- Coding a client “on the fly”
 - Creating a CMTS
 - Creating a CableModem
 - Running the CableModem.
- Simulating hundreds of thousands of devices in 2'
 - Creating 1 CMTS
 - Creating “n” CableModems (different vendors !)
 - Powering on everything and analyzing results.
- ICMP : It’s already implemented...so... Can we ping it ?
YEAH !
- Creating other network devices

TOOLS



Scapy 

DHCP Server

TFTP Server



Protocol Buffers

CONCLUSIONS



Simulate network traffic (real/custom)



Crafting/implementing custom devices / protocols is easy



Recreate expensive scenarios.



Stress testing



Bug hunting, crafting your own packets and flows

THANK YOU !

