

UPV/EHU

BBK

IRTF - NFVRG

Rethinking NFV: Supporting Efficient Packet Processing

Eduardo Jacob

<Eduardo.Jacob@ehu.eus>



I E T F®

IETF 100 Singapore
November 11-17, 2017

i2t Research Group

erren ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

CAMPUS OF
INTERNATIONAL
EXCELLENCE

Agenda

- NFV as we know it.
- Evolution of the technology
- ETSI POC #43
- Limitations
- Some pointers to solution?

- University of the Basque Country
 - Public University of the Basque Country Autonomous Region (Spain) with about 4.500 teachers, 45.000 students and lectures in 112 degree courses in 83 topics.
 - Distributed University: 3 campuses in three provinces.
 - The research group belongs to the Department of Communications Engineering (90 people, 38% full time researchers) and is located in the Faculty of Engineering of Bilbao.
 - The department is running several EU H2020 research projects.

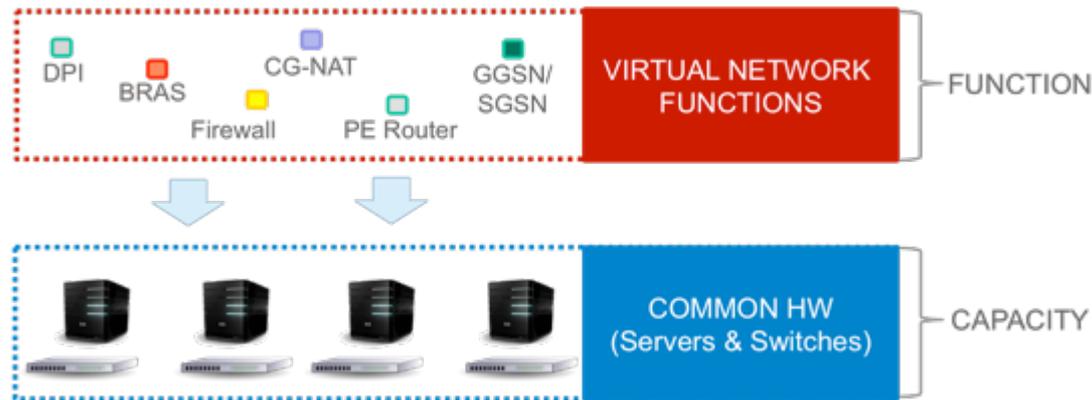


i2t Research Group

<http://i2t.ehu.eus>

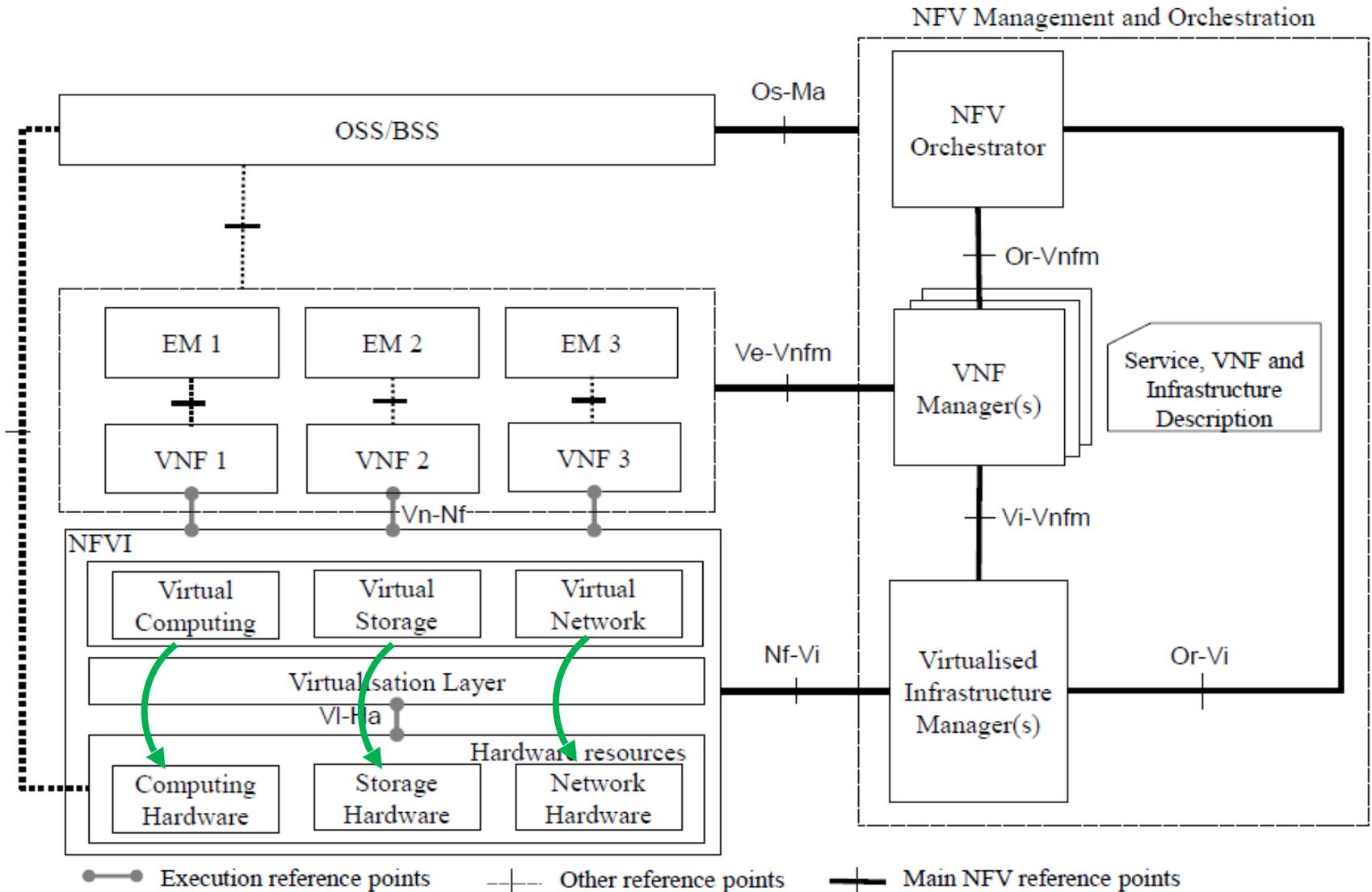


- NFV: Network Function Virtualization
 - Replacing dedicated equipment by commodity computing hardware (originally x86)



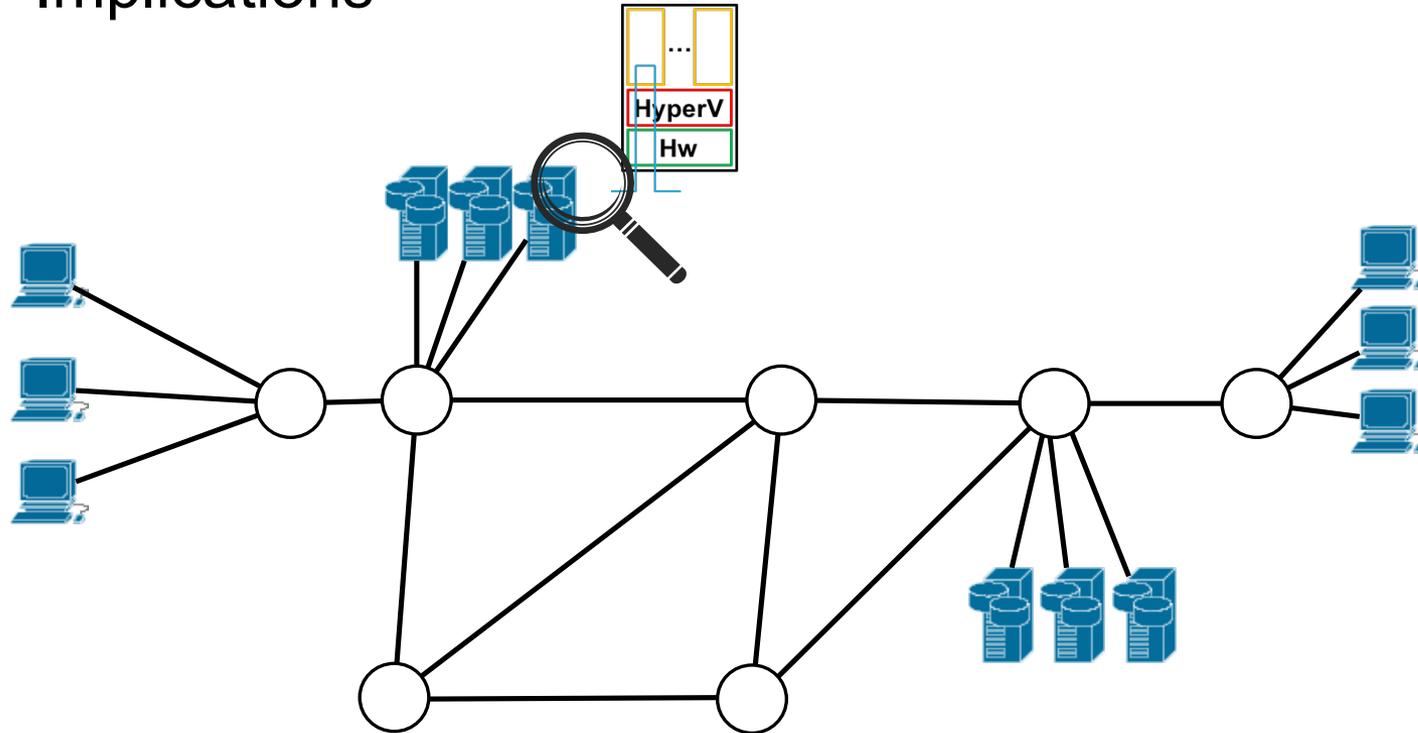
- This meant going to cloud technologies
 - Hypervisors
 - Manipulating packets in user space
- Originally network was static
 - Later it could “configured” or defined as needed (SDN, in broad sense)





NFV as it should be

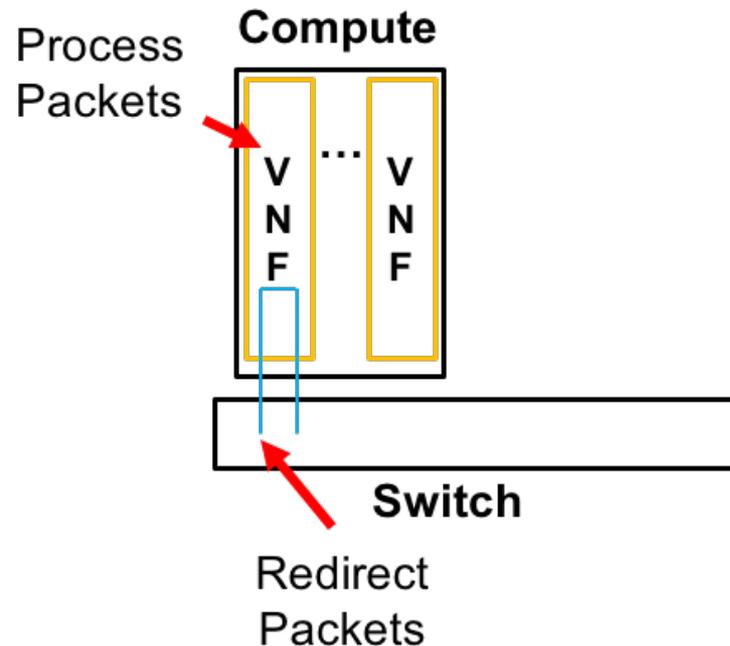
- Implications



- Packet processing implies having a compute node
 - Generic packet processing in any point of the network is not possible.



- Can we get a more efficient packet processing?
- In classic NFV approach
 - Compute node treats the packet
 - Network Element (Switch) reroutes packets



For now **VNF**,
means some
kind of CPU
based
processing

- There are two competing approaches targeting a more efficient packet processing.
 - General purpose CPU process
 - Enhanced dataplanes.
- With the added complications of dataplanes implemented in general purpose CPUs...
- Some (old?) ideas

	Price/port	Environmental tolerance	Features Adaptability	Backplane bandwidth	Processing Speed
Computing boxes	↑↑↑	↑	↑↑↑	↑	↑
Switches	↑	↑↑↑	↑	↑↑↑	↑↑↑



- Processing on the dataplane (an overview...)
- OF based stateless processing (not considering meters, group ports...)
 - ASIC manufactured for third party switch manufacturers:
 - I.e.: Broadcom Trident II
 - Similar features on switches
 - Vendor specific ASICs
 - I.e.: Aruba Provision
 - Specific features (custom pipelines)
 - NPU based
 - I.e.: Noviflow
 - FPGA based
 - I.e.: Corsica
 - X86 based dataplane
 - I.e.: xdpd, Openvswitch, cpqd switch
 - ...
- Stateful processing (with different approaches)
 - OpenState[1] mealy finite state machines (FSM)
 - OpenPacket[2] extended finite state machines (XFSM)
 - FAST (Flow-level State Transitions [3])
 - P4 [4]

Many of these are in research or experimental status !!!

[1] <https://qmonnet.github.io/whirl-offload/2016/07/17/openstate-stateful-packet-processing/>

[2] <https://qmonnet.github.io/whirl-offload/2016/11/09/open-packet-processor/>

[3] M. Moshref, A. Bhargava, A. Gupta, M. Yu, and R. Govindan, "Flow-level state transition as a new switch primitive for SDN," in 3rd workshop on Hot topics in software defined networking,

[4] <https://p4.org>



- Packet processing on the computing node (an overview...)
- Processing is done with x86 code.
- Stateful processing.
- Concept of execution environment shrinking to improve processing speed, boot/setup time...
 - Hypervisor + VM
 - I.e.: XEN, VMware
 - Containers
 - Docker, LXC, LXD
 - Unikernels
 - ClickOS[5], Mirage [6]
- Low level packet processing improvements
 - Mostly based on bypassing operating system's TCP/IP stack.
 - Many integrate in other tools like OpenStack
 - In kernel processing
 - XDP[7] (eXpress Data Path)
 - User space
 - NetMAP[8]
 - DPDK
 - Snabb[9]

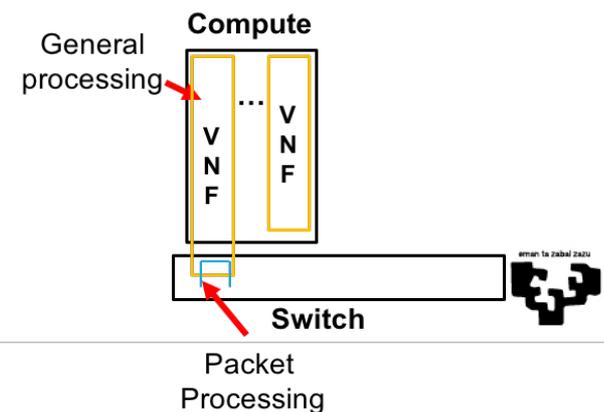


- CPU evolution?
 - Over speed/cores...
 - Hybrid approaches
 - Intel CPU+FPGA
 - Not clear if this usable at packet processing applications: DataCenter.

[5] <http://cnp.neclab.eu/projects/clickos/>
 [6] <https://mirage.io>
 [7] <https://www.iovisor.org/technology/xdp>
 [8] <https://github.com/luigirizzo/netmap>
 [9] <https://github.com/snabbco/snabb>

- Conclusions

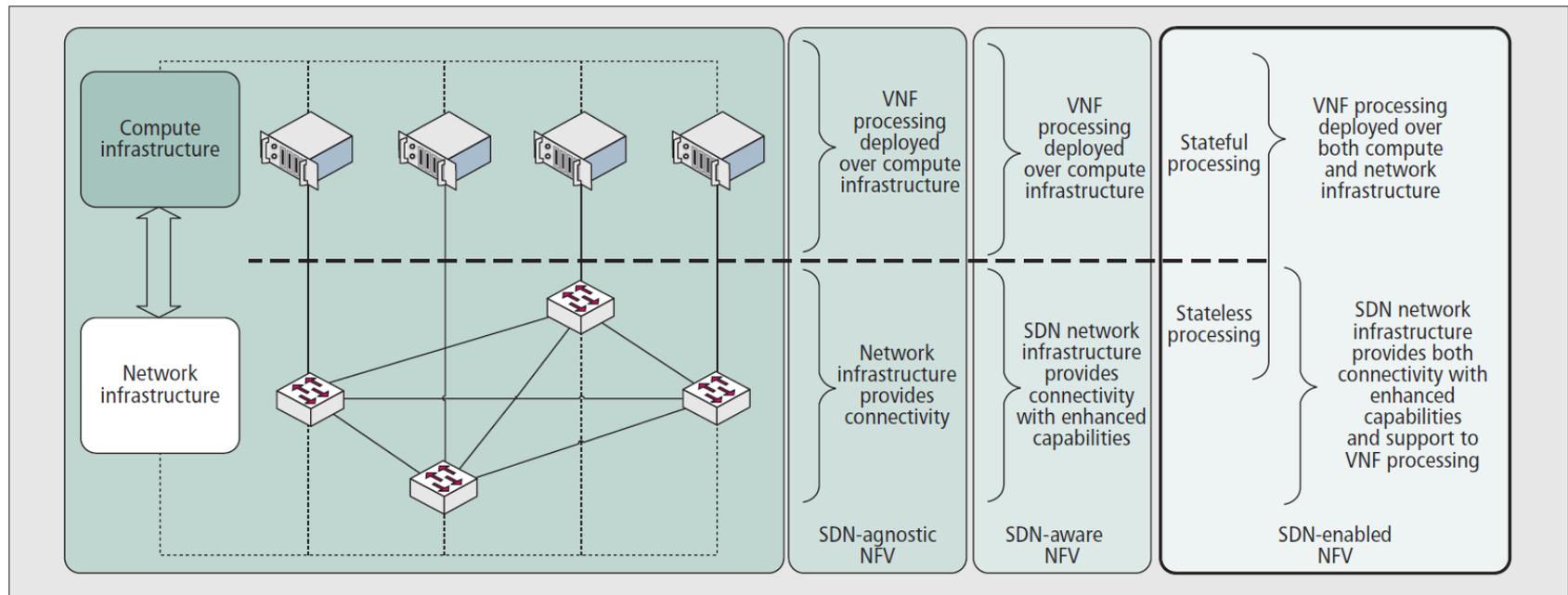
- Boundaries between computer and switch are no longer equivalent to data and packet management ability.
- A more subtle difference involves “state”
 - Traditional switches are stateless
 - Some new players involve stateful solutions (on both silicon and soft switches)
- There are champions on each specialty
 - Classical VNF are the kings of stateful processing
 - Full x86 code support and (much) memory.
 - Classical switches are queens of stateless processing.
 - OpenFlow switches could be considered 1st class players in stateless processing..
- Is there a place for an Hybrid VNF?
 - Combined processing in a general purpose CPU and in the switch?



- Why should we care?
 - There is place for improving current architectures
 - Let's see ETSI POC #43: Towards an efficient Data Plane processing



- This started some years ago...
- SDN-Enabled NFV concept
(now it would have another name)

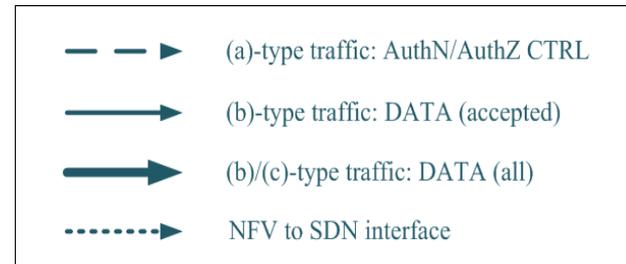
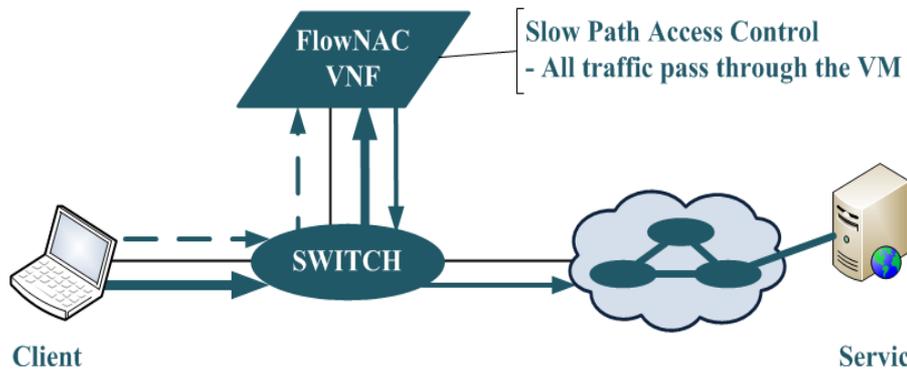


[10] "Toward an SDN-enabled NFV architecture" IEEE Communications Magazine (April 2015)

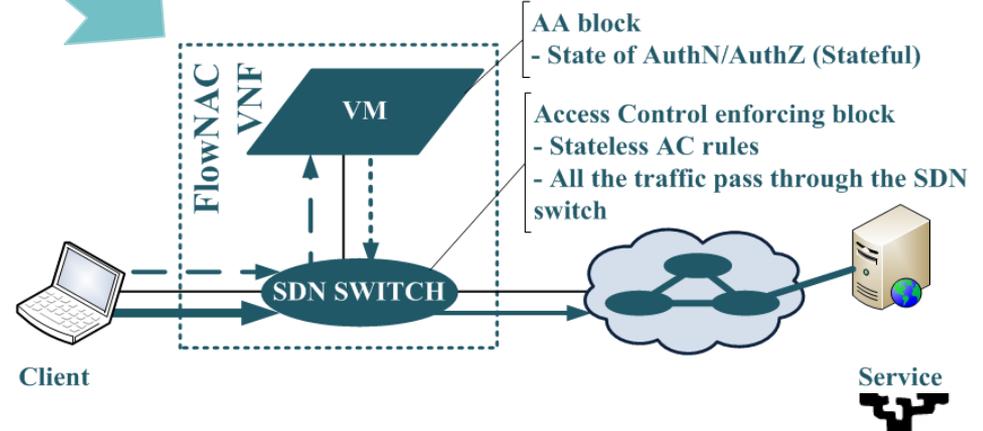


- Use case VNF: FlowNAC, a Flow aware Network Access Control

SDN-agnostic NFV / SDN-aware NFV



SDN-enabled NFV

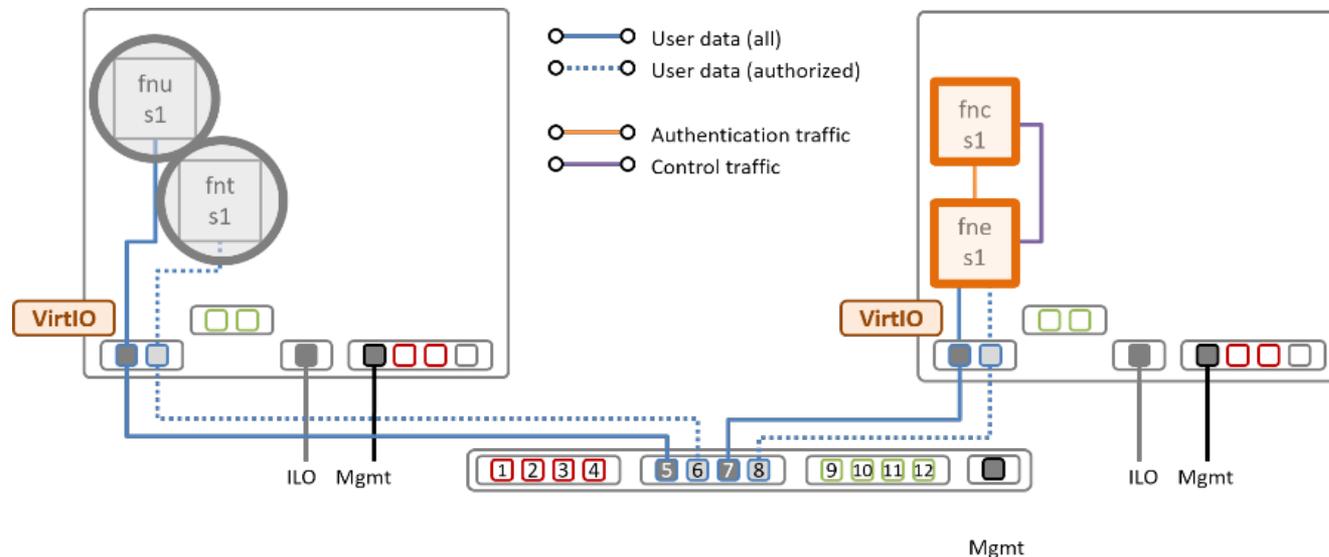


- ETSI PoC#43 Towards an efficient Data Plane processing.
 - Three scenarios tested and compared:



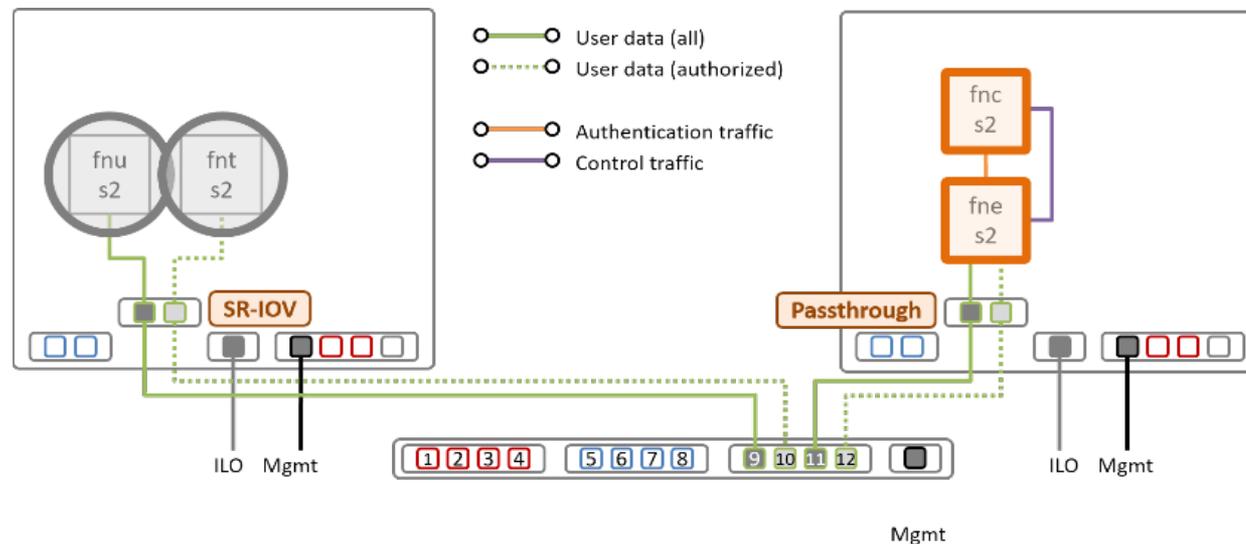
- ETSI PoC#43 Towards an efficient Data Plane processing.
 - Three scenarios tested and compared:

Classical VM based processing



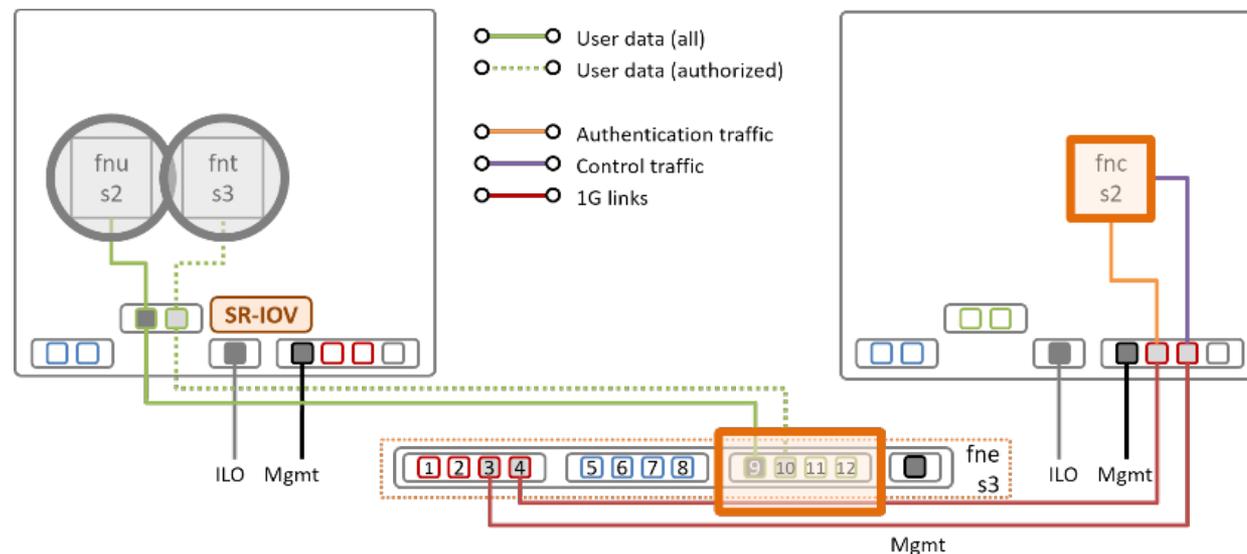
- ETSI PoC#43 Towards an efficient Data Plane processing.
 - Three scenarios tested and compared:

Classical VM with EPA

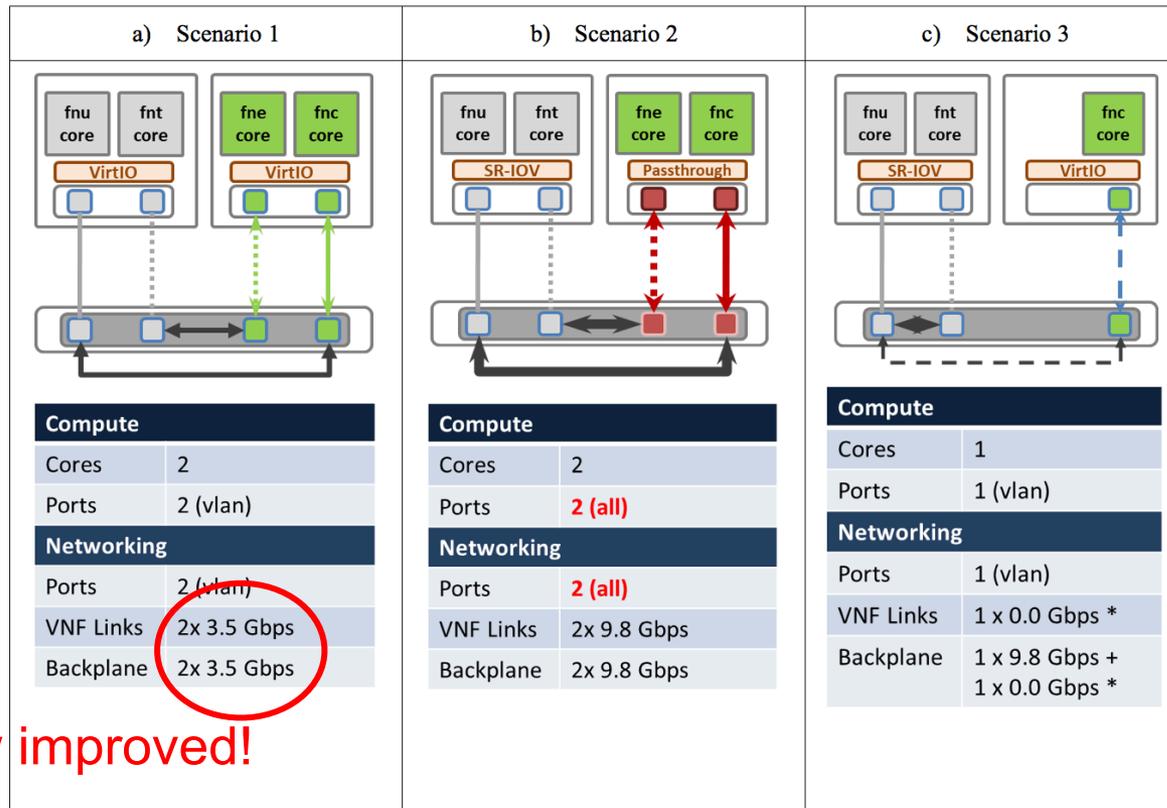


- ETSI PoC#43 Towards an efficient Data Plane processing.
 - Three scenarios tested and compared:

Stateless Processing offloaded to OF Switch



ETSI PoC#43 Towards an efficient Data Plane processing.



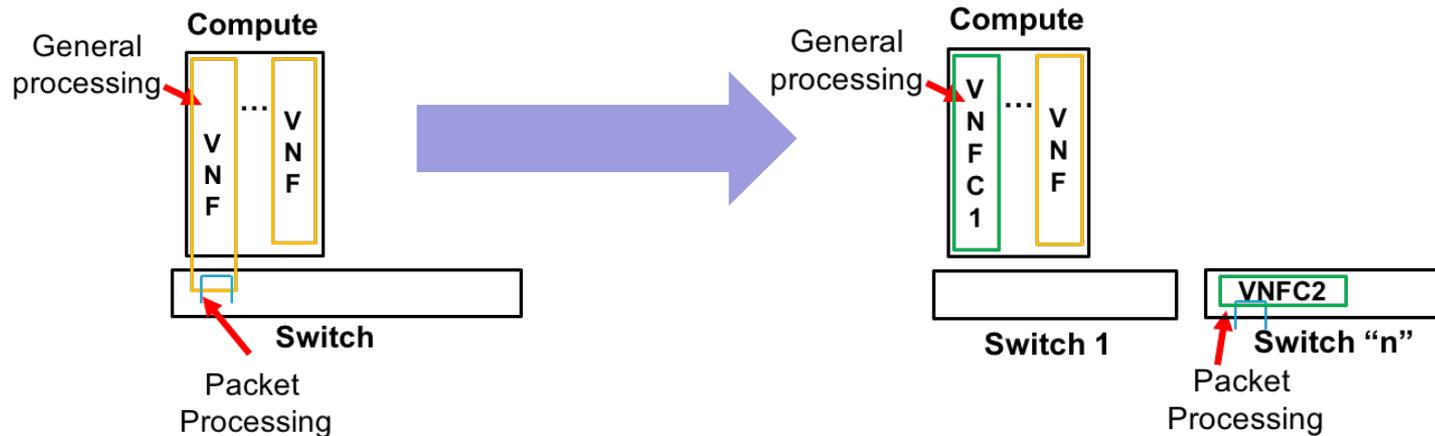
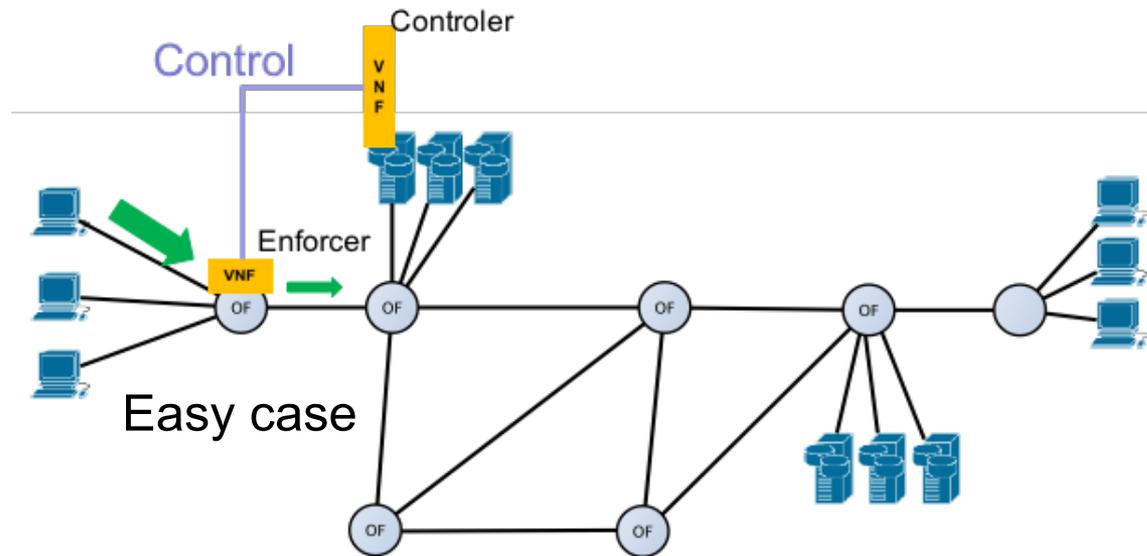
Now improved!

Comparison of resource usage in the three scenarios

- The traffic doesn't leave the data plane.
- Enforcing done at full switch speed.
- The control channel is almost no used: less that 5kb per re/authentication.
- This means that the control function can get topologically decoupled from the enforcing point.
- The policy can be enforced in any switch.
- A core is freed.



- A “split” VNF (or a VNF with two VNFC)
 - But one of them on a switch

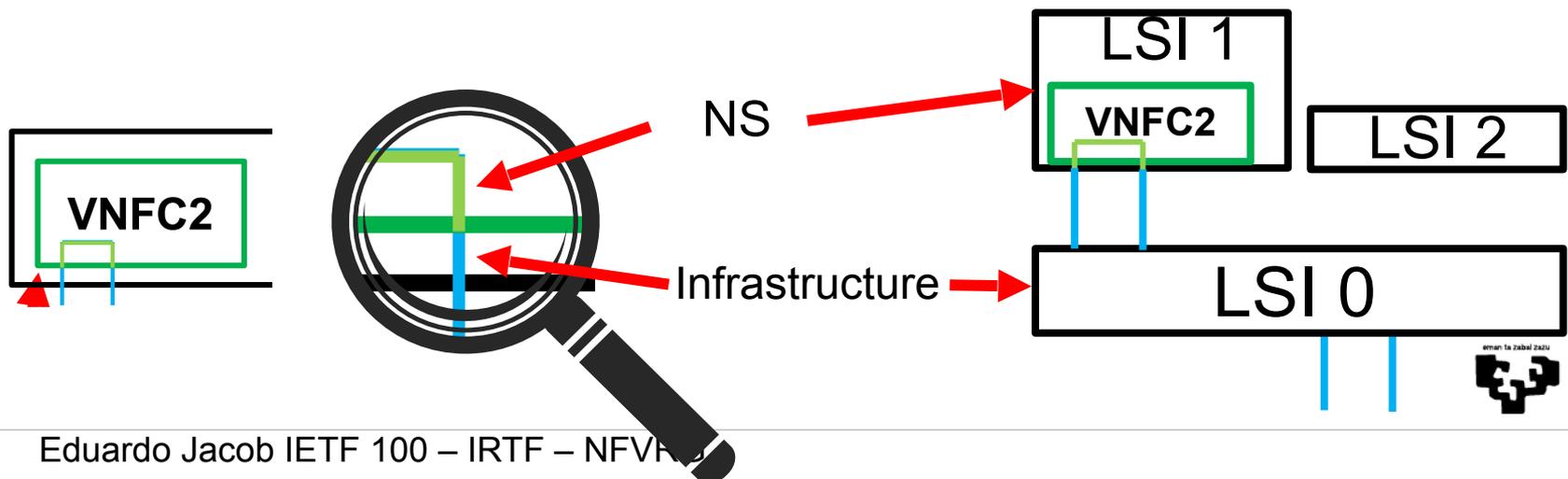


- Limitations

- The NFV+SDN equation is usually written as:

NFV + SDN

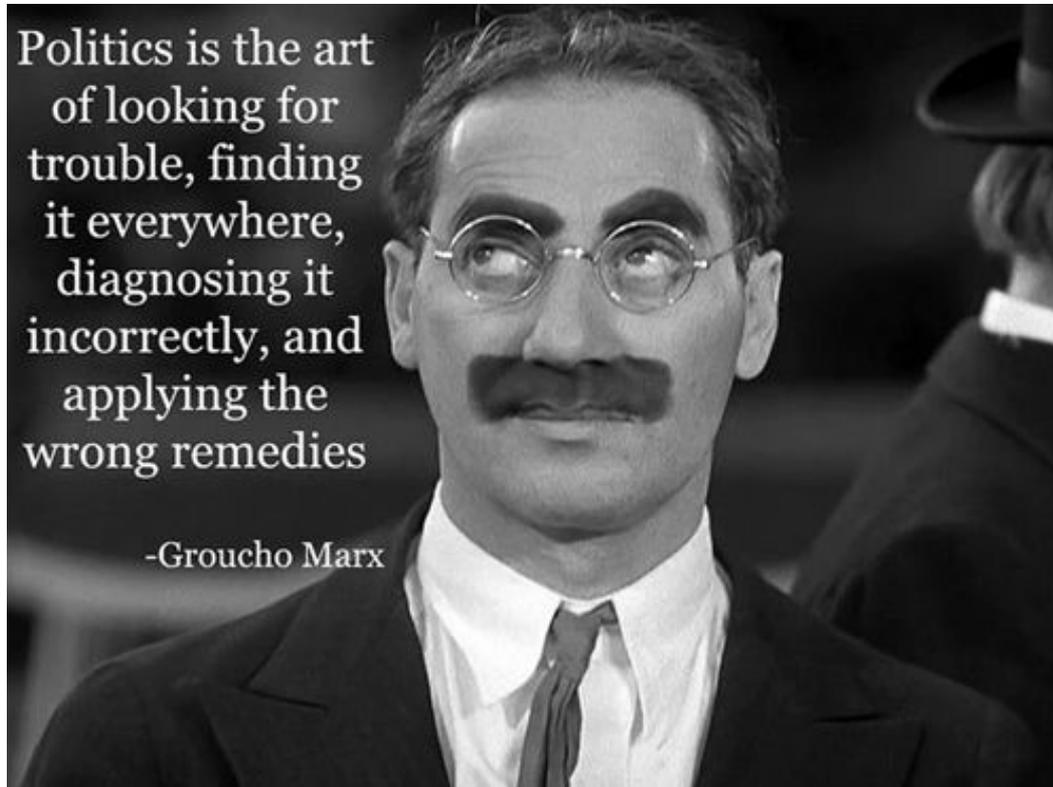
- SDN was incorporated later.
- The network was not considered to be handled by the Network Service (user)
- Delegating control of part of the network equipment is not easy (not only technically, but also administrative, ie OF instance)



- Limitations (II)

- Network tends to reorganize: self-healing... supposing packets are not processed: You enter the “placement” game in a delay bounded playground (good luck)
- VNFD (VNF descriptor) and NSD (NS descriptor) do not contemplate processing outside the computing node.
- There are too many options/formats/descriptions for packet processing operation (there is no “x86 machine” - like code to describe it, well there could be some): It’s not about discussing the format (VM/Container/Unikernel...) in which code is delivered.

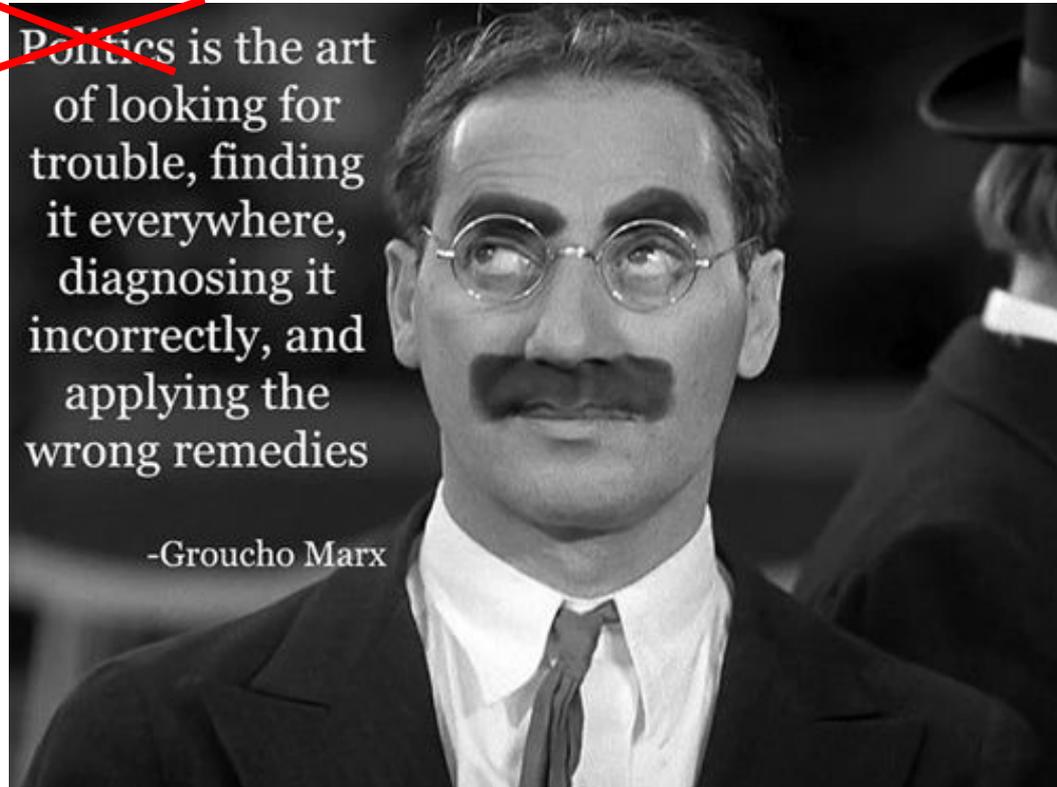
- Pointers to a solution? (not a closed list)
 - Rewrite the equation as NFV+SDN.
 - Consider the network really part of the software involved in the NS provisioning
 - You could get tunneling or cyphering as part of the link description and have it provisioned on any network node.
 - You could ease the deployment of advanced slicing mechanisms.
 - A NS should be able to ask for a network connection that could include properties like VLAN translation, tunneling, cyphering or resilience and let the SDN controller take care of it.
 - Network equipment should be able to “virtualize” its resources and delegate control to third parties.
 - Consider recursivity.
 - It’s more than an EPA (Enhanced Platform Awareness) issue over a special HW.
 - Get some coffee...



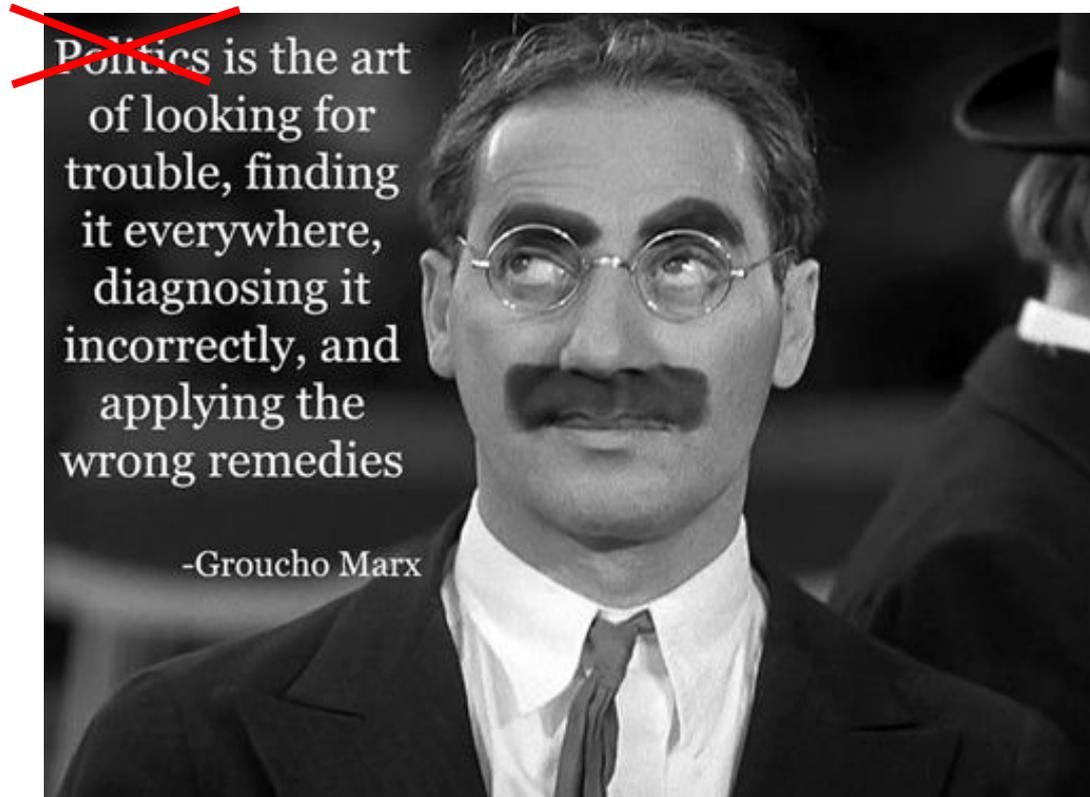
Research

~~Politics~~ is the art
of looking for
trouble, finding
it everywhere,
diagnosing it
incorrectly, and
applying the
wrong remedies

-Groucho Marx



Research



Let's hope not!!!



Thank you!

Eduardo Jacob



CAMPUS OF
INTERNATIONAL
EXCELLENCE