# An NSH-Enabled Architecture for Virtualized Network Function Platforms

Vinícius F. Garcia, Leonardo da C. Marcuzzo, Giovanni V. Souza, Lucas Bondan, Jéferson C. Nobre, Alberto E. Schaeffer-Filho, Carlos R. P. dos Santos, Lisandro Z. Granville, Elias P. Duarte Jr.

# Summary

- **Introduction**
- **Related Works**
- **Platform Architecture**
- **Prototype**
- **Evaluation**
  - Experimental Setup
  - Experimental Results
  - Results Overview
- **Conclusion**

# Introduction

- **Traditional Networks**
  - Physical appliances (middleboxes)
  - Lack of life cycle operations flexibility
  - High CApital and OPerational Expenditures (CAPEX and OPEX)
- **Network Function Virtualization (NFV)**
  - Network paradigm
  - Decoupling network functions from its associated hardware
  - Virtualization technologies (virtual machines,  containers)
  - Software plane flexibility

# Introduction

- **Virtualized Network Function (VNF)**
  - Block of the ETSI NFV architecture
  - Composed by two elements:
    - Network Function (NF)
    - VNF Platform
- **Virtualized Network Function Platform**
  - Environment that supports the execution a NF
  - Use many NFV enablers (*e.g.* operating systems, packet accelerator, programming languages)

# Introduction

- **Service Function Chain (SFC)**
  - Sophisticated services
  - Multiple network functions connected
- **Network Service Header (NSH)**
  - SFC traffic steering protocol
  - Encapsulate L3 packets
  - IETF SFC architecture (classifier, forwarder, proxy)
  - NSH aware and NSH unaware

# Introduction

CURRENT VNF PLATFORMS ARE NOT CREATED USING STANDARDIZED ARCHITECTURES

# Introduction

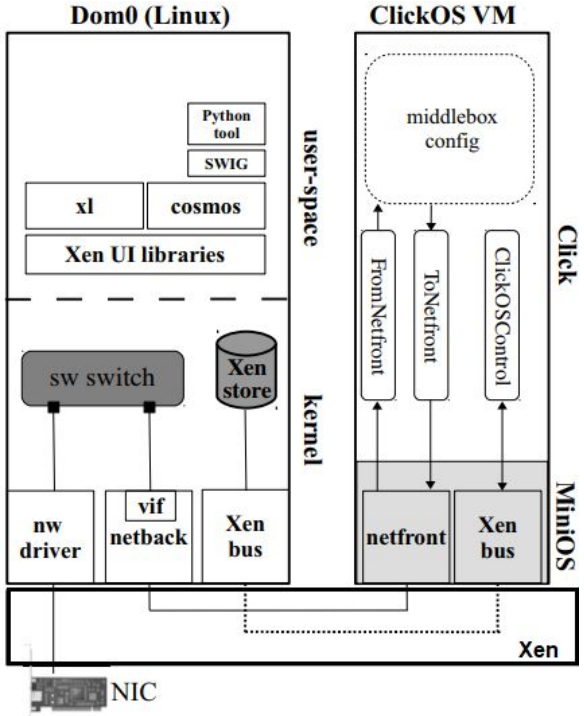
CURRENT VNF PLATFORMS ARE NOT CREATED USING STANDARDIZED ARCHITECTURES

NETWORK SERVICE HEADER IS NOT NATIVELY PROCESSED IN THE VNF PLATFORMS

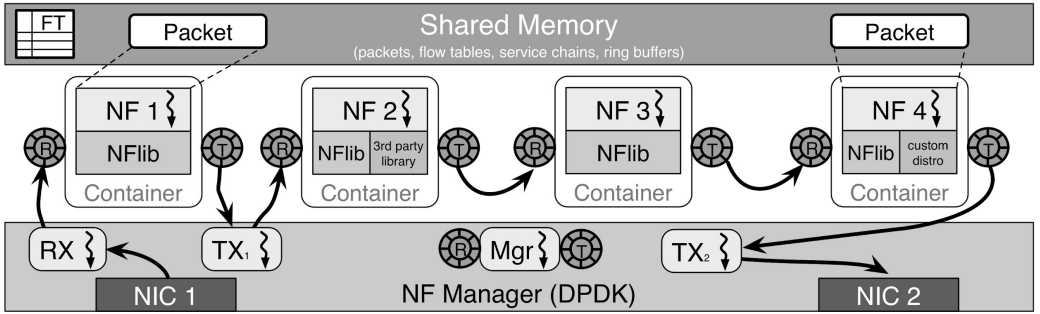
# Objective

**INTRODUCE A COMPREHENSIVE ARCHITECTURE FOR VNF PLATFORMS THAT STRICTLY ADHERES TO ETSI REQUIREMENTS AND PROVIDES SUPPORT FOR NSH**

# Related Works




ClickOS Platform Architecture




OpenNetVM Platform Architecture

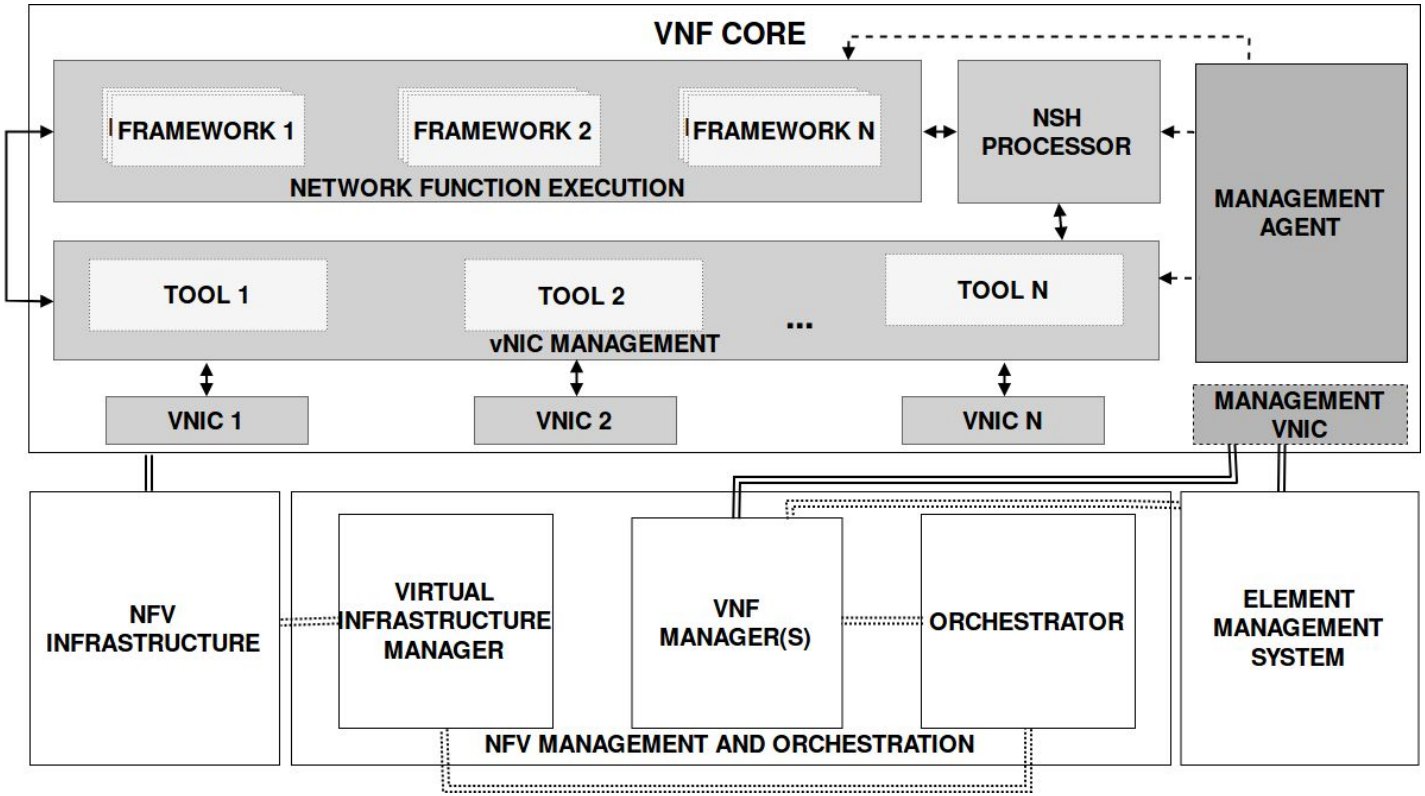
# Related Works

- **ClickOS and OpenNetVM do not have standardized platforms**
  - Do not support Network Service Header
  - Monolithic implementations
    - It is not possible to expand the enablers set
      - NFLib is not supported by ClickOS
      - Click Modular Router is not supported by OpenNetVM
  - Some VNF requirements are not addressed
    - ClickOS Xen Hypervisor dependency (integration)
    - OpenNetVM containers restrictions (portability)

# Architecture

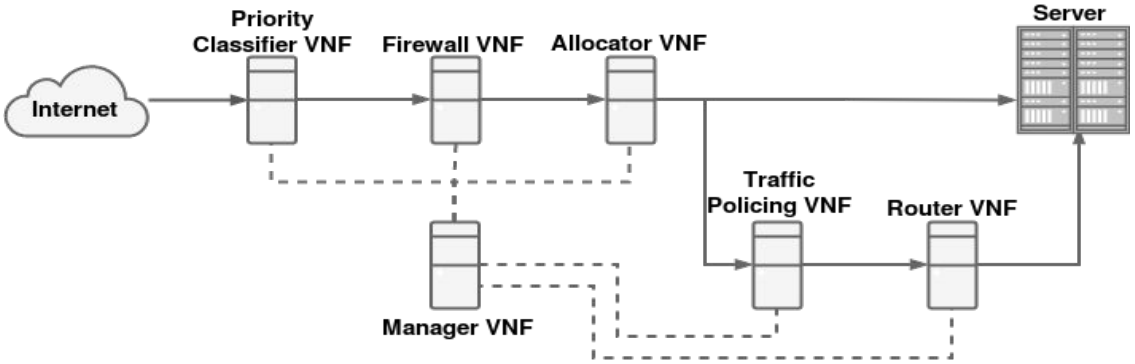
- **Flexible architecture based on internal modules**
  - Minimalist OS (Unikernel or Container)
  - Few main internal modules
    - vNIC Management
    - NF Execution
    - VNF management and statistics collector
    - NSH Processor (Optionally)
  - Internal modules can be changed according to the scenario
    - Support for existing NFV Enablers
    - API for management of new developed modules

# Architecture

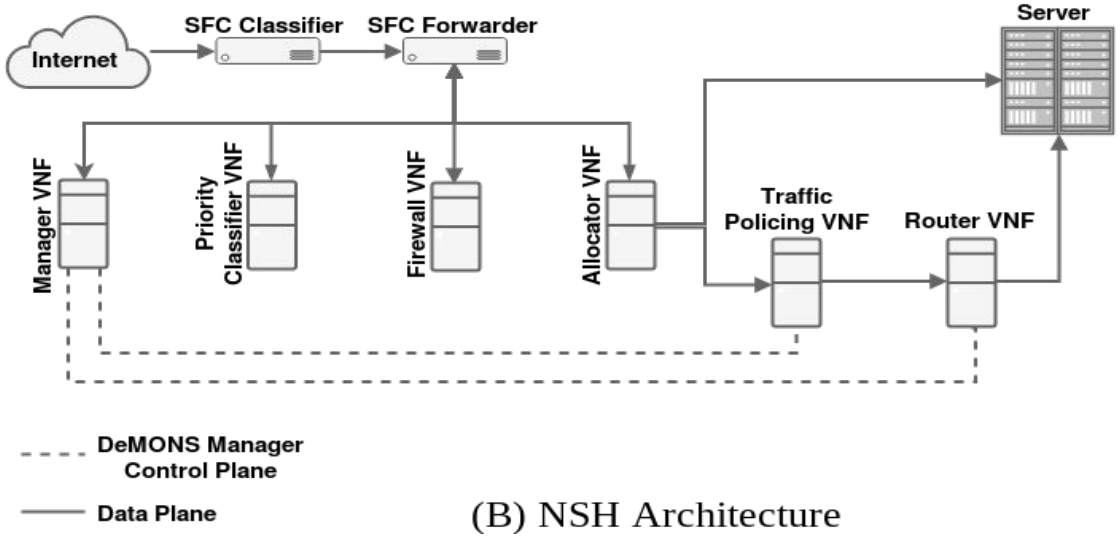


# Prototype

- **vNIC Management**
  - DPDK
  - L2 Sockets
- **Network Function Execution**
  - Click Modular Router
  - Python 3
- **Management Agent**
  - RESTful Web Services




(A) Non-NSH Architecture




(B) NSH Architecture

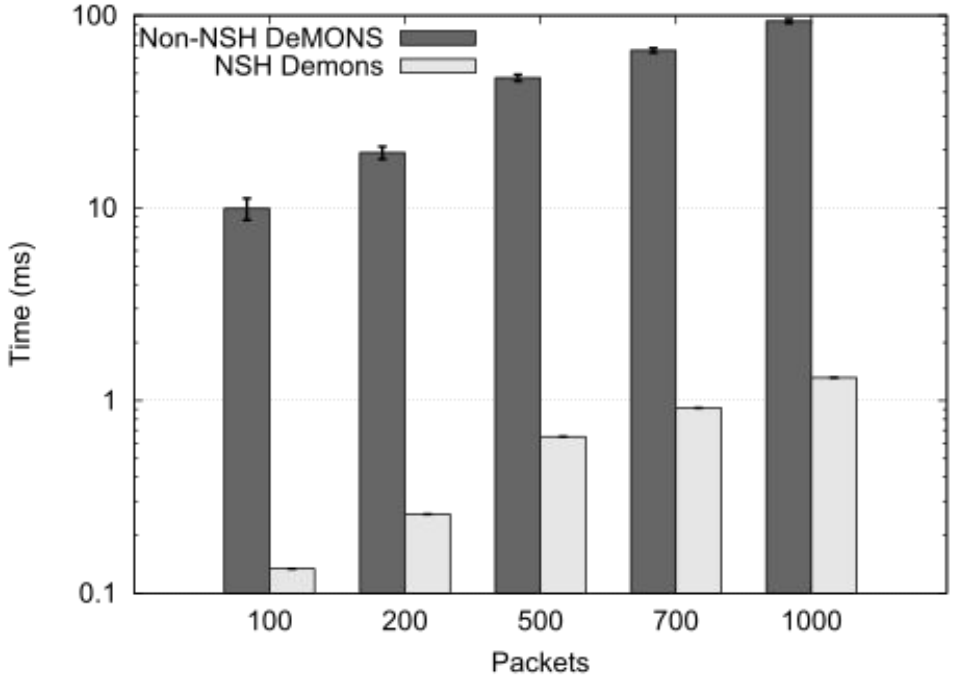
# Experimental Setup

- **DeMONS**
  - NFV solution for DDoS Mitigation
- **Non-NSH Architecture**
  - Every packet is processed by all VNFs
- **NSH Architecture**
  - IETF Architecture
  - In-band control
    - Context Header
    - Service Index

# Experimental Setup

- **Non-NSH DeMONS**
  - Non-NSH DeMONS uses an UDP Socket to retrieve the flow reputation from the Manager VNF
- **NSH DeMONS**
  - NSH DeMONS uses the NSH Context Header field to retrieve the flow reputation from the packet itself
- **Expected Results**
  - The in-band control leads to significant differences in terms of processing time (in favor to NSH-based)
  - Local access rather than remote access

# Experimental Results




Reputation Retrieval Aggregated Time

# Conclusion

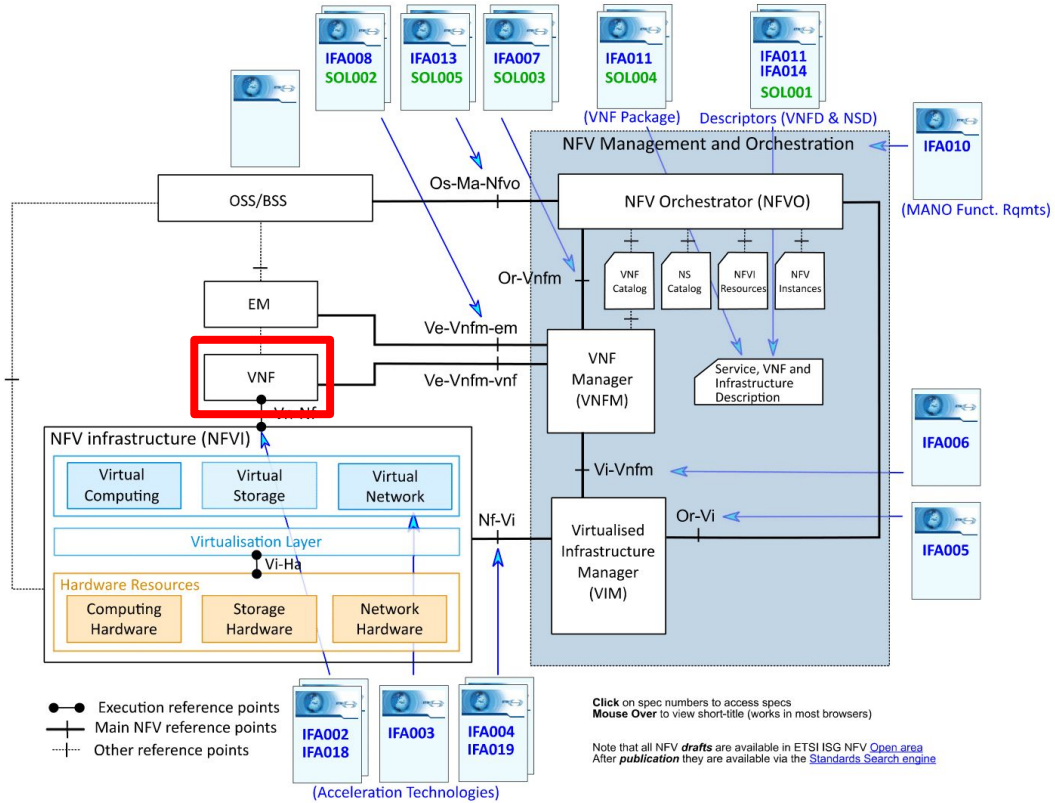
- **VNF Platforms Architecture**
  - Native NSH processing
  - Support to different NFV enablers
  - Flexible and modular architecture
  - Platform prototype
- **Future Works**
  - NSH investigation on current description models (*e.g.* TOSCA)
  - Platform prototype evolution
    - Other packet processing frameworks support (*e.g.* VPP)
    - Packet accelerators new options (*e.g.* netmap)

# An NSH-Enabled Architecture for Virtualized Network Function Platforms

## Thanks!!

vfulber@inf.ufsm.br

# Questions


IFA008
SOL002
IFA013
SOL005
IFA007
SOL003
IFA011
SOL004
(VNF Package)
IFA011
IFA014
SOL001
Descriptors (VNFD & NSD)
IFA010
(MANO Funct. Rqmts)
NFV Management and Orchestration
OSS/BSS
Os-Ma-Nfvo
NFV Orchestrator (NFVO)
Or-Vnfm
VNF Catalog
NS Catalog
NFVI Resources
NFV Instances
EM
Ve-Vnfm-em
VNF
Ve-Vnfm-vnf
VNF Manager (VNFM)
Service, VNF and Infrastructure Description
IFA006
Vi-Vnfm
NFV infrastructure (NFVI)
Virtual Computing
Virtual Storage
Virtual Network
Nf-Vi
Or-Vi
Virtualisation Layer
Vi-Ha
Virtualised Infrastructure Manager (VIM)
IFA005
Hardware Resources
Computing Hardware
Storage Hardware
Network Hardware
Execution reference points
Main NFV reference points
Other reference points
IFA002
IFA018
IFA003
IFA004
IFA019
(Acceleration Technologies)
Click on spec numbers to access specs
Mouse Over to view short-title (works in most browsers)
Note that all NFV drafts are available in ETSI ISG NFV Open area
After publication they are available via the Standards Search engine

# Questions

**Experimental Setup (Physical Machine) ->**

- Intel Core i7-4790K@3.60Ghz
- 8GB RAM DDR4
- Debian 8
- KVM Hypervisor