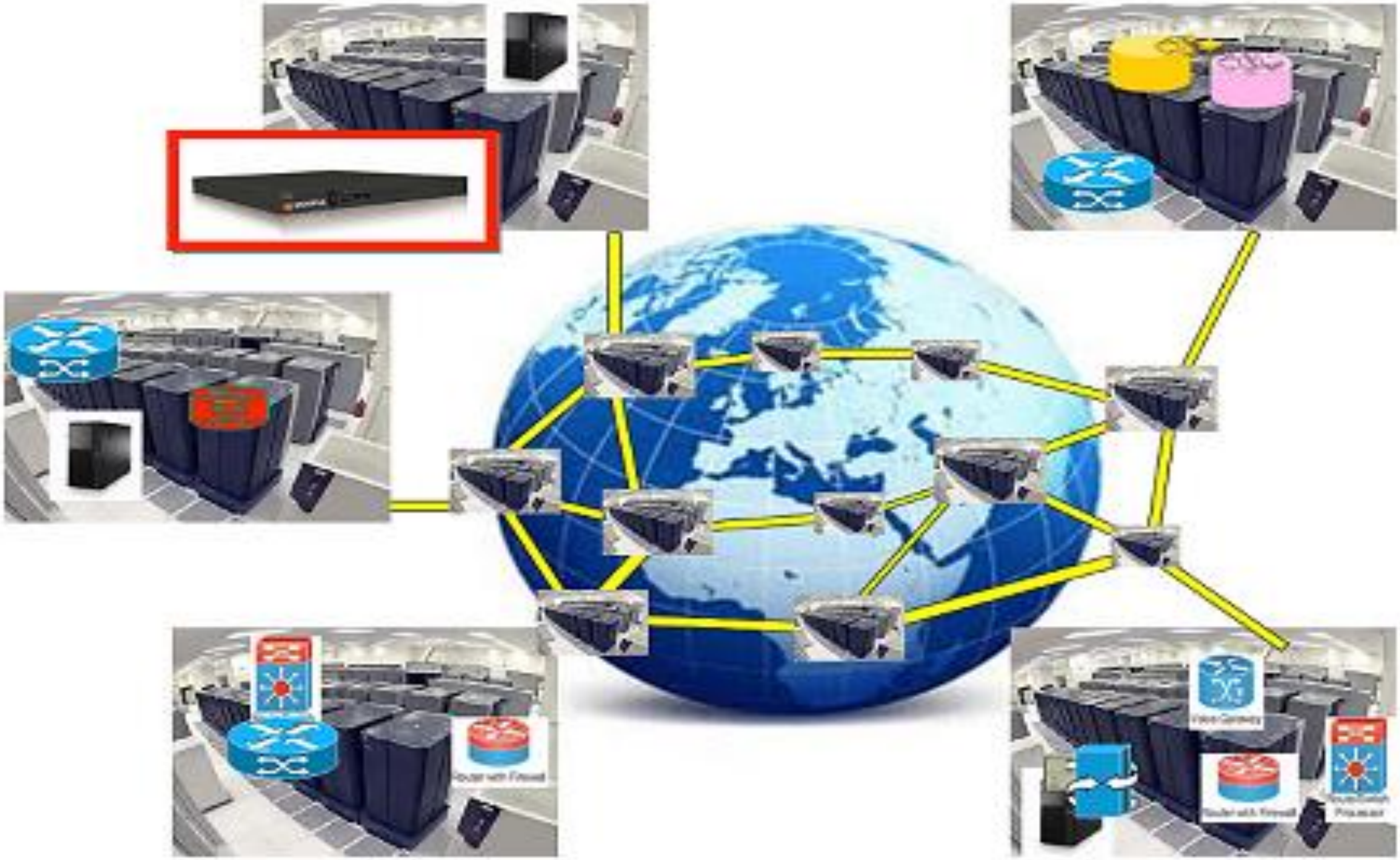
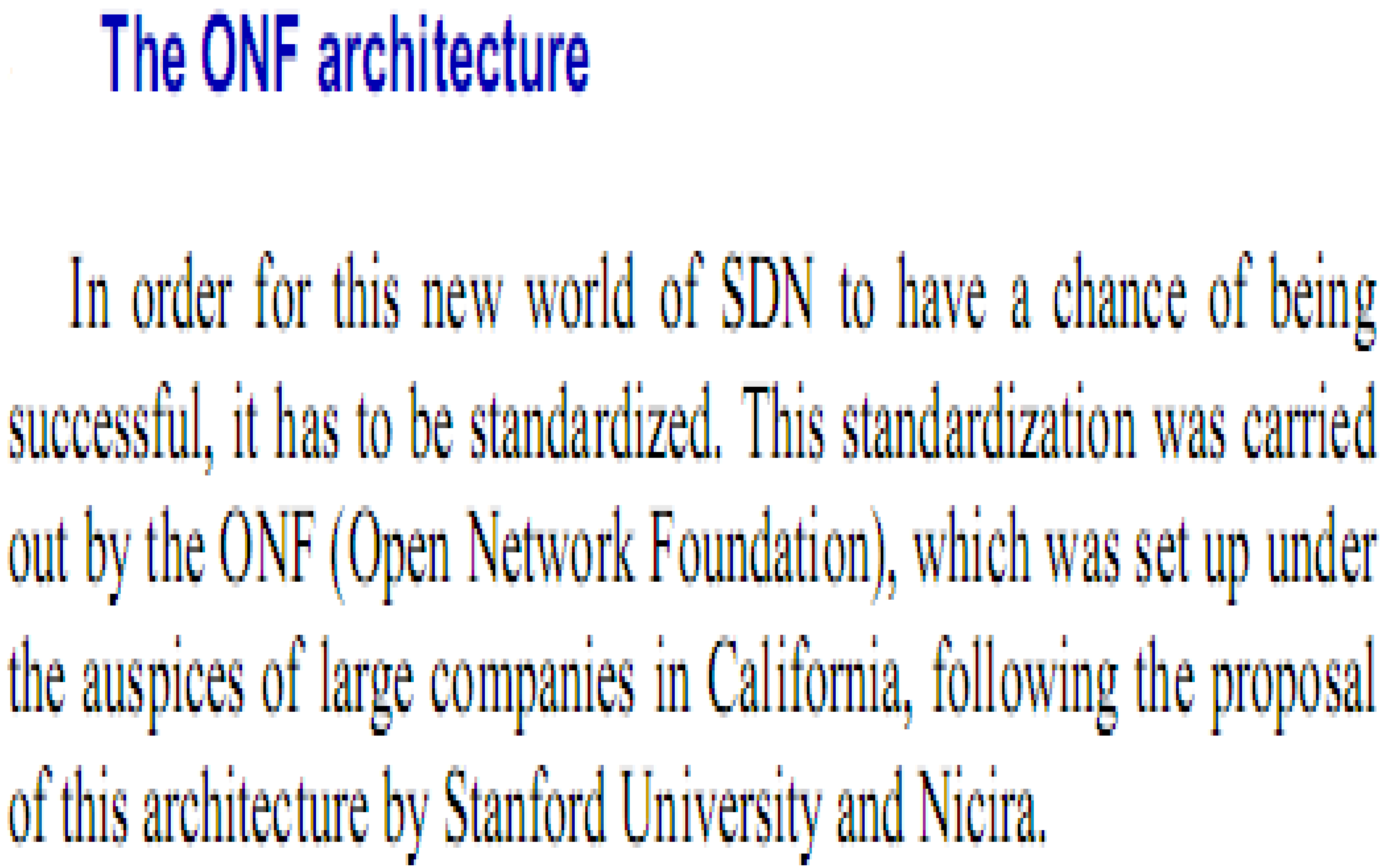
Chapitre III :

OPNFV : Open NFV



A view of SDN networks of tomorrow. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip

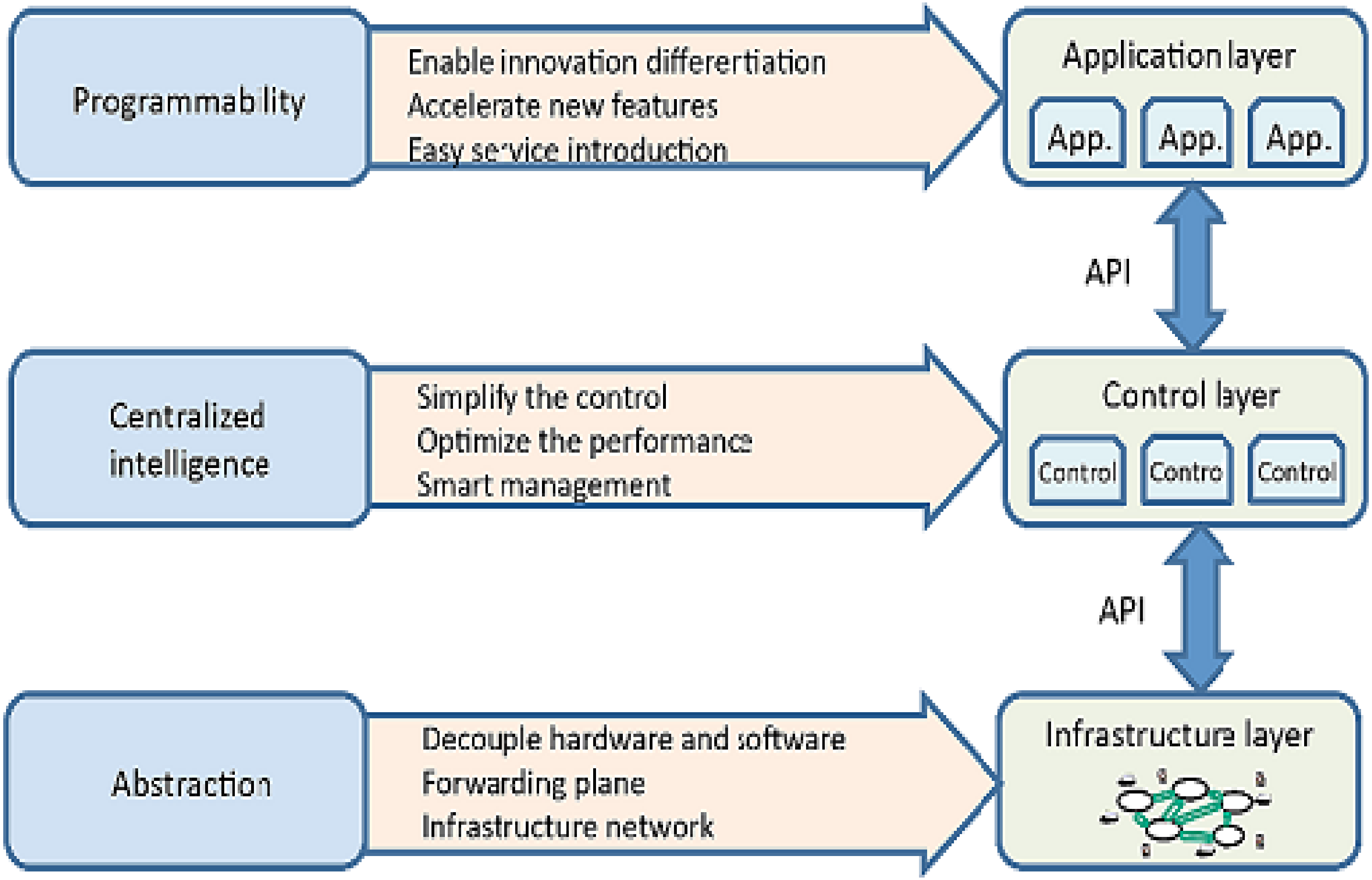
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



The ONF architecture

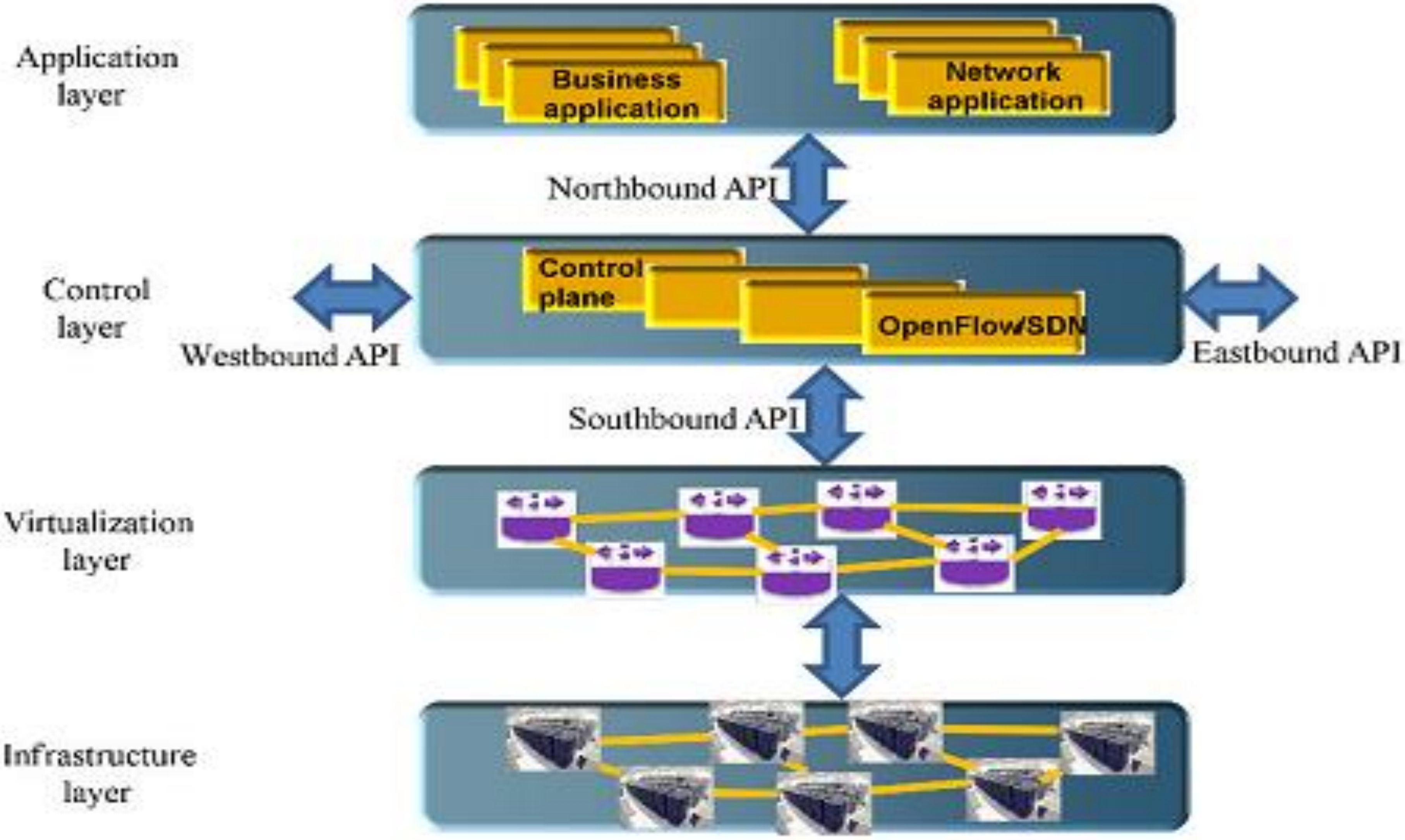
In order for this new world of SDN to have a chance of being successful, it has to be standardized. This standardization was carried out by the ONF (Open Network Foundation), which was set up under the auspices of large companies in California, following the proposal of this architecture by Stanford University and Nicira.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



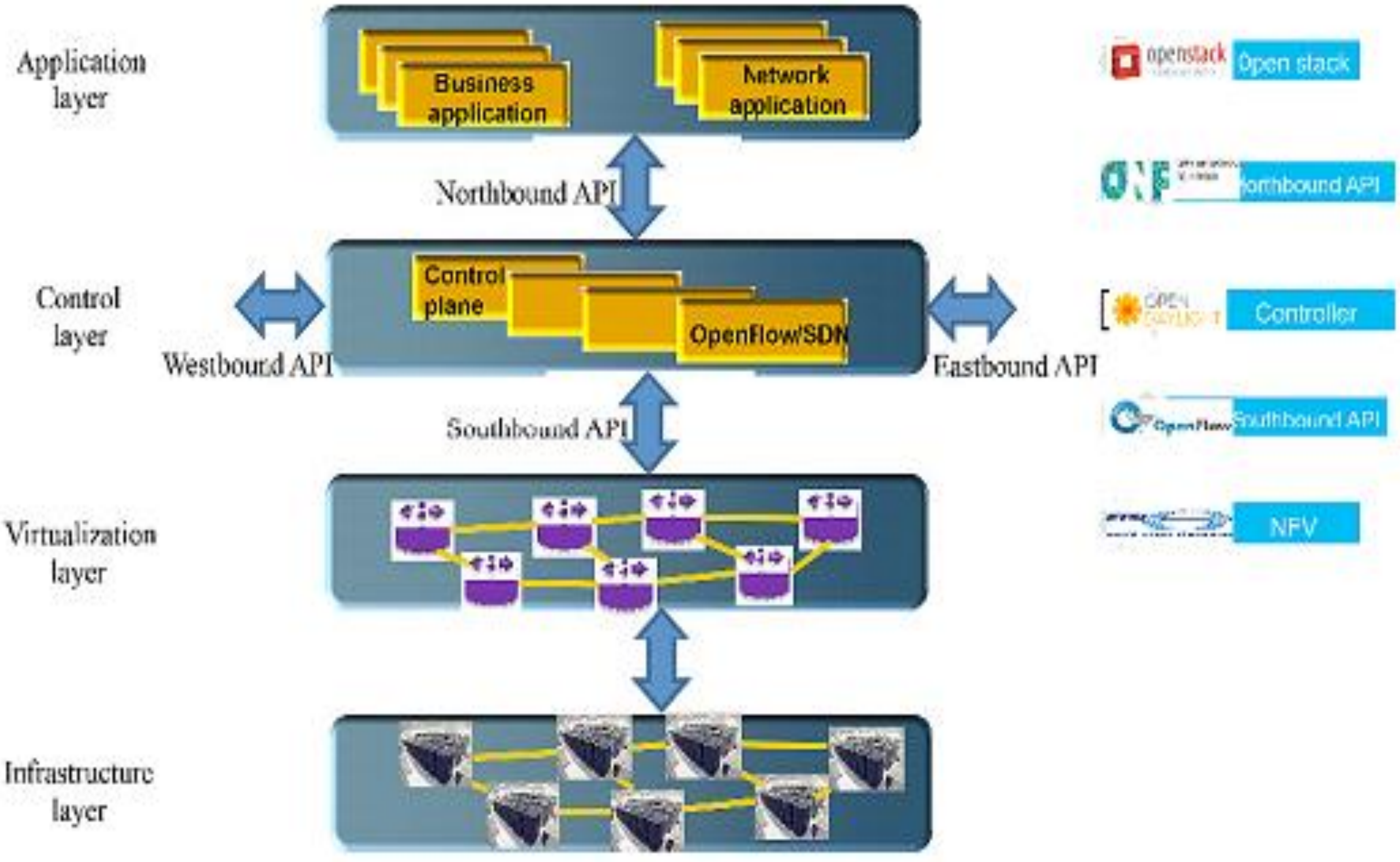
The ONF architecture

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



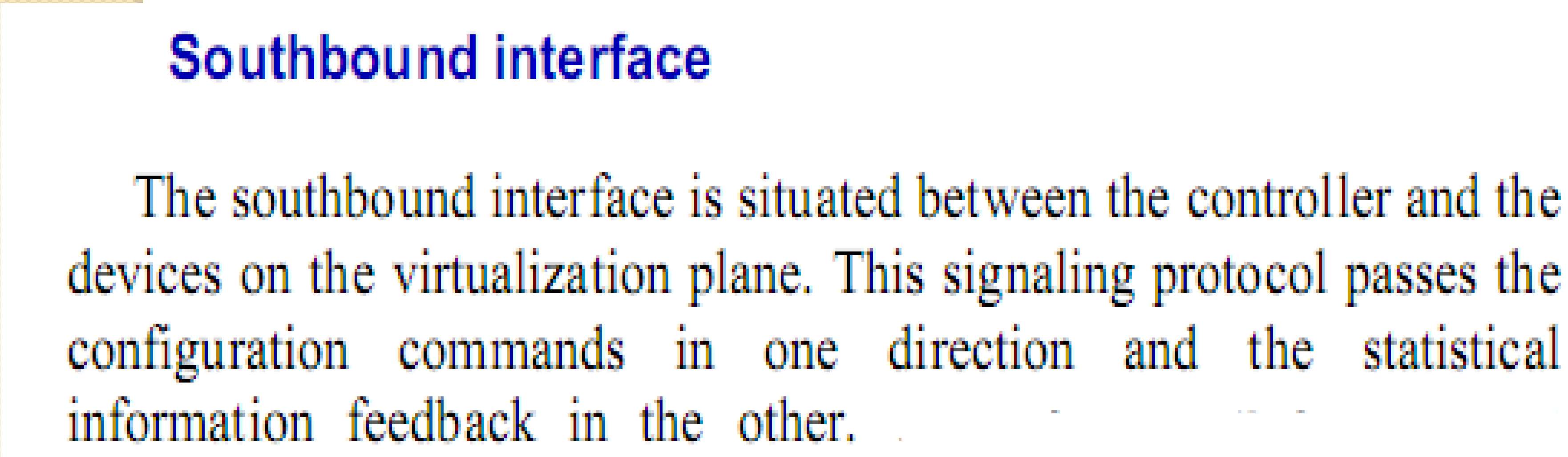
The SDN architecture

Source : **Guy Pujolle,** Software Networks:Virtualization, SDN, 5G and Security, First Edition, 2015.



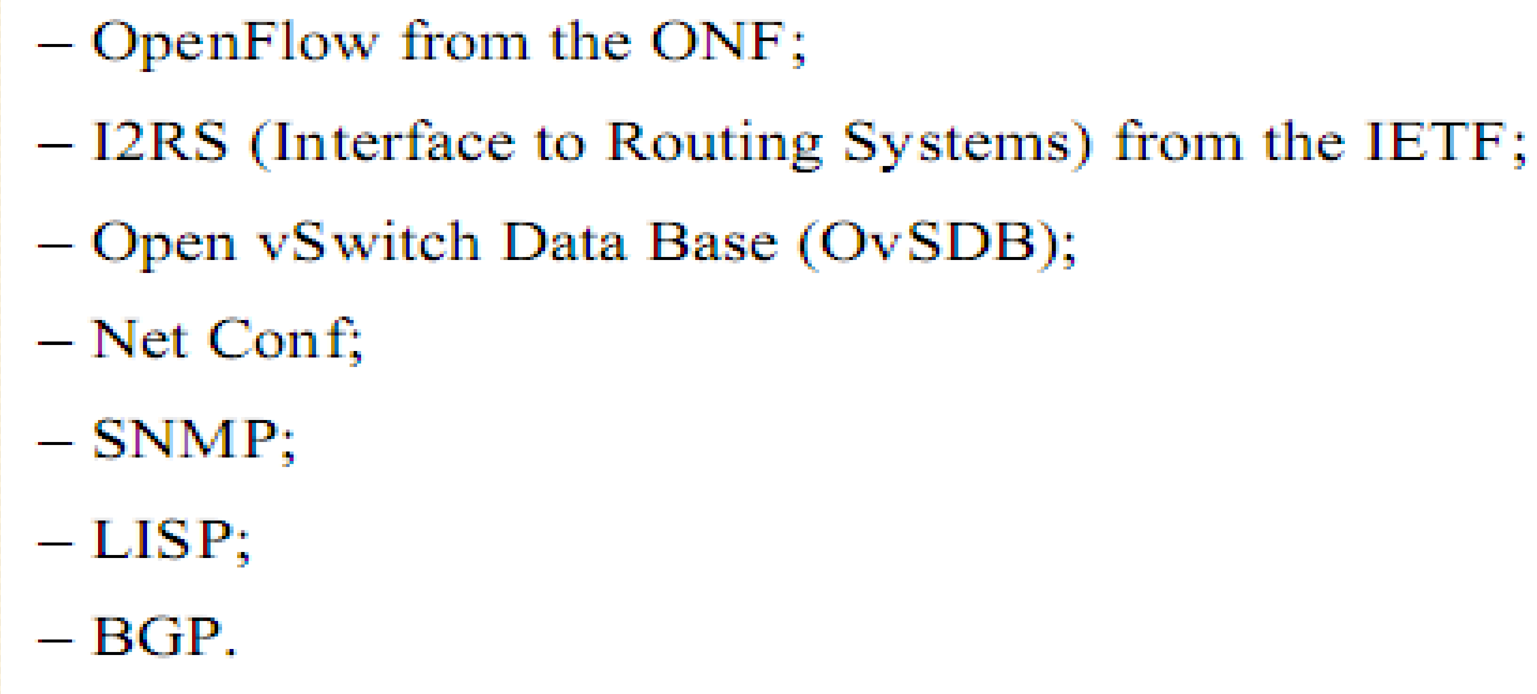
Example of Open Source developments

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Southbound interface

The southbound interface is situated between the controller and the devices on the virtualization plane. This signaling protocol passes the configuration commands in one direction and the statistical information feedback in the other.



– OpenFlow from the ONF;

– I2RS (Interface to Routing Systems) from the IETF;

– Open vSwitch Data Base (OvSDB);

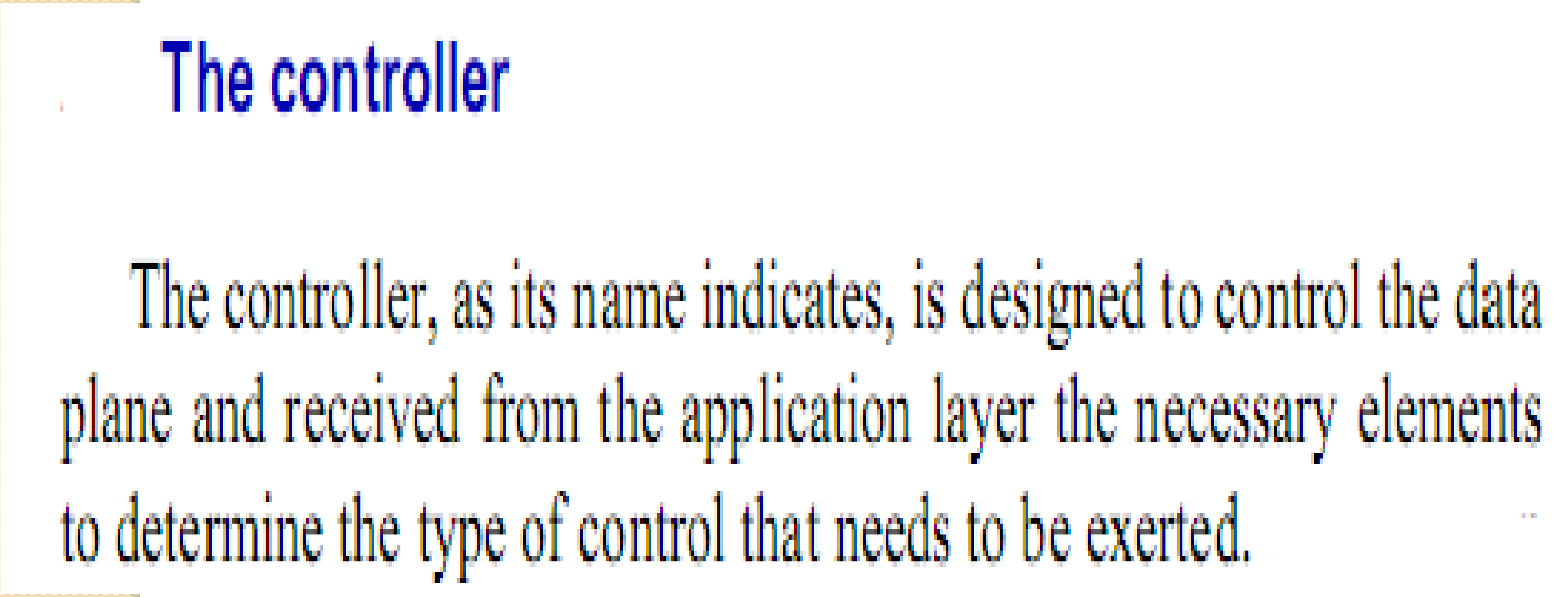
– Net Conf;

– SNMP;

– LISP;

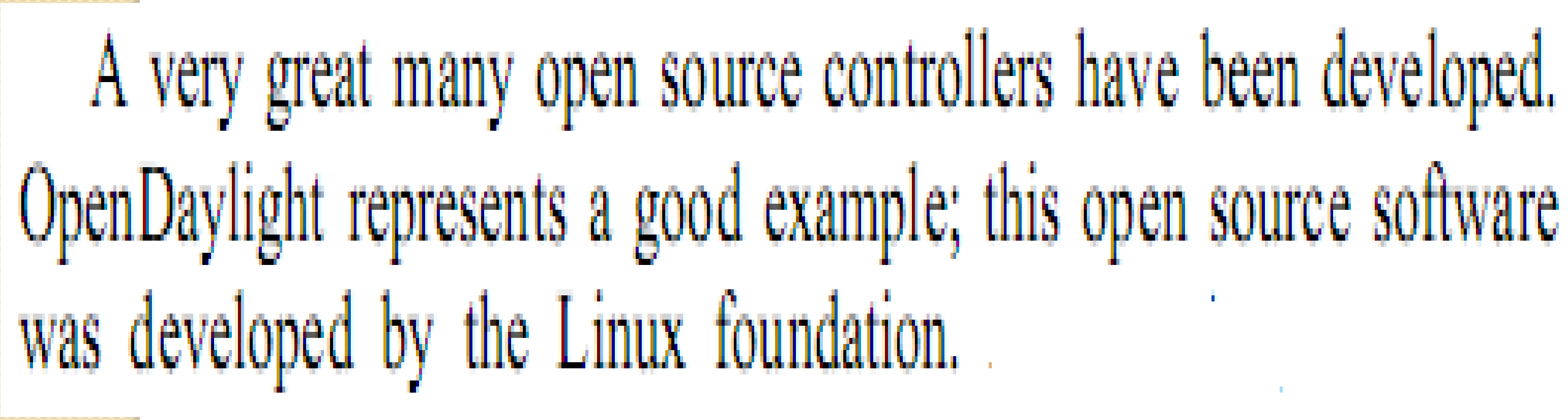
– BGP.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



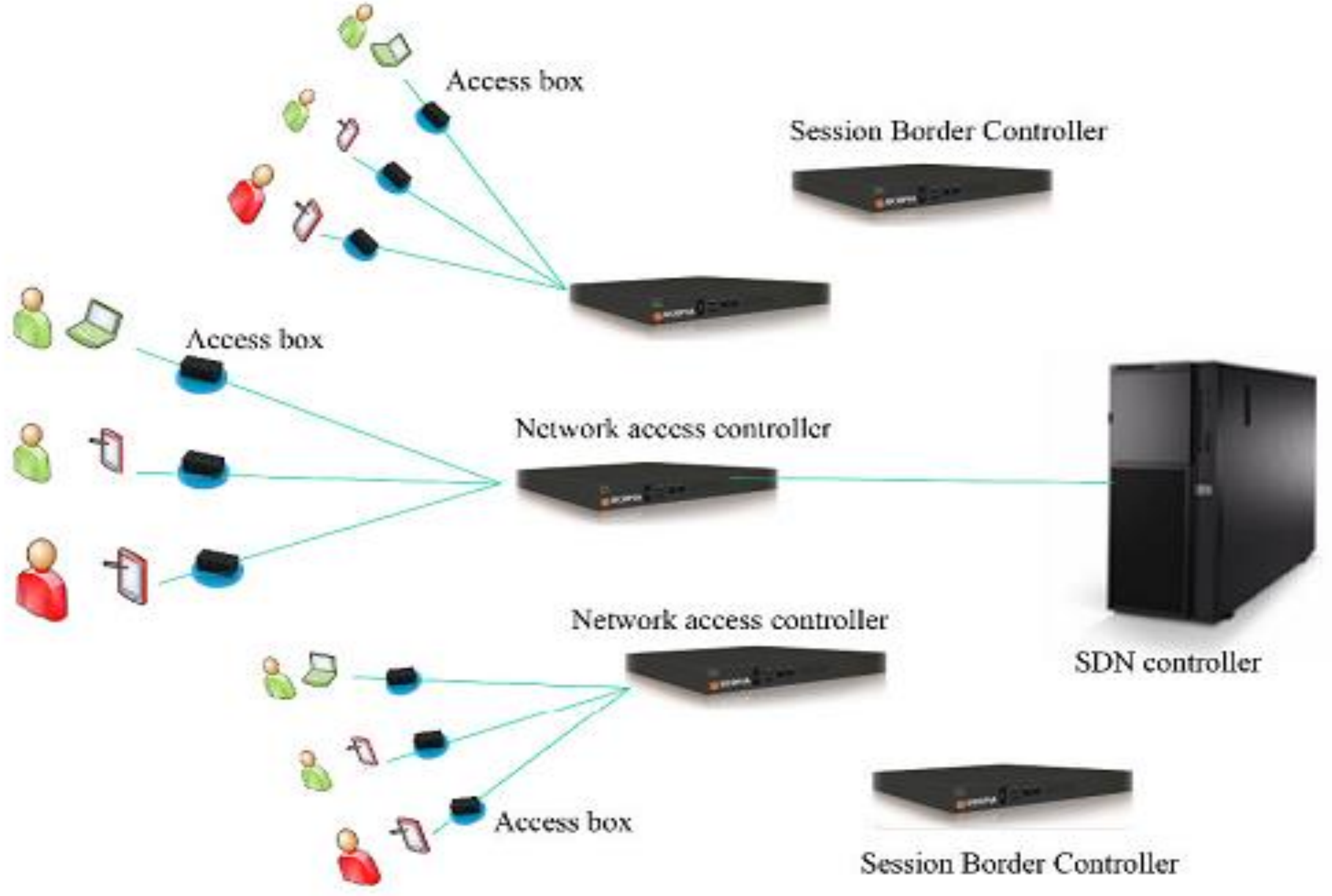
The controller

The controller, as its name indicates, is designed to control the data plane and received from the application layer the necessary elements to determine the type of control that needs to be exerted.



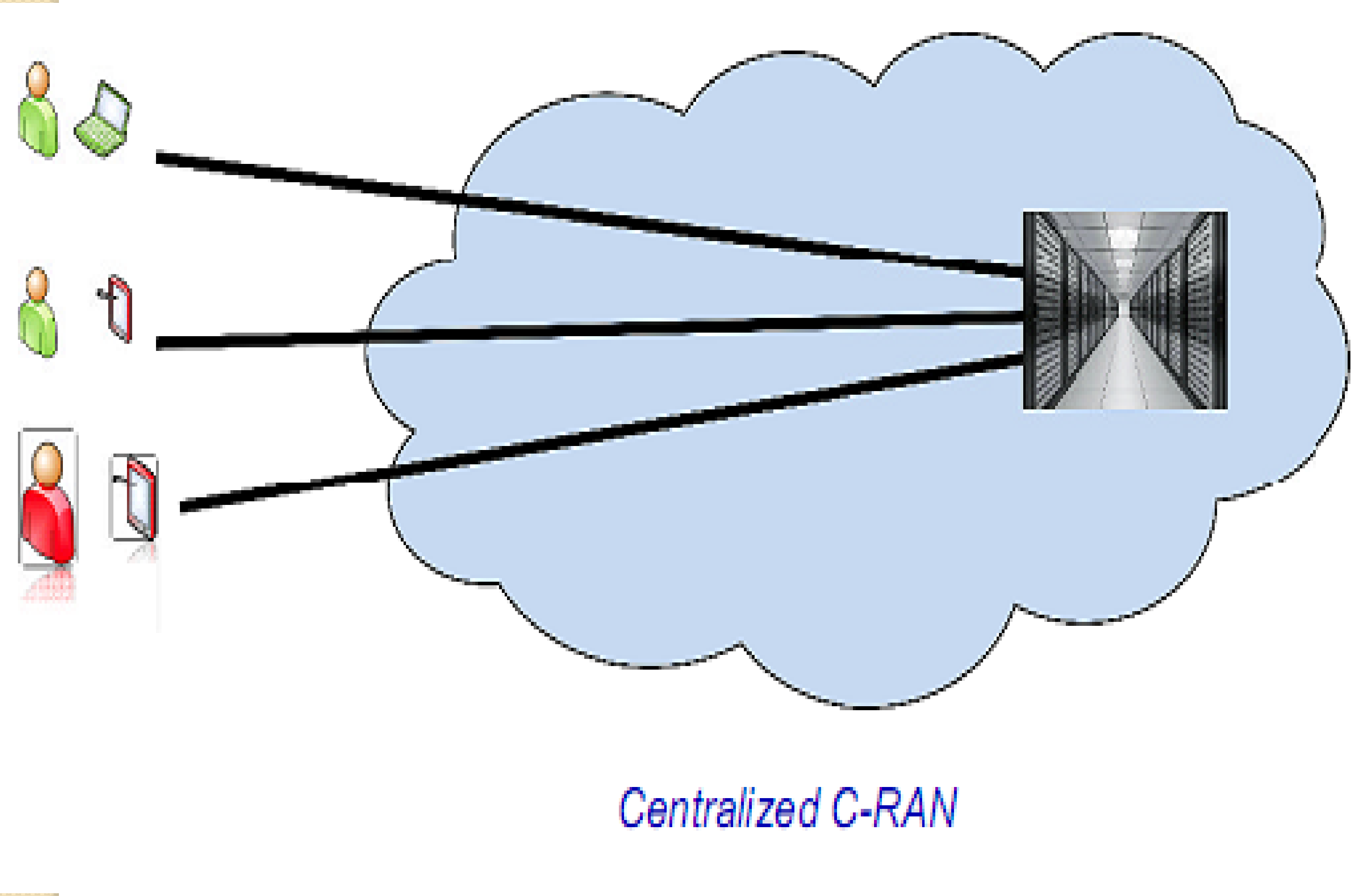
A very great many open source controllers have been developed. OpenDaylight represents a good example; this open source software was developed by the Linux foundation.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.


Access box
Session Border Controller
Access box
Network access controller
Network access controller
SDN controller
Access box
Session Border Controller

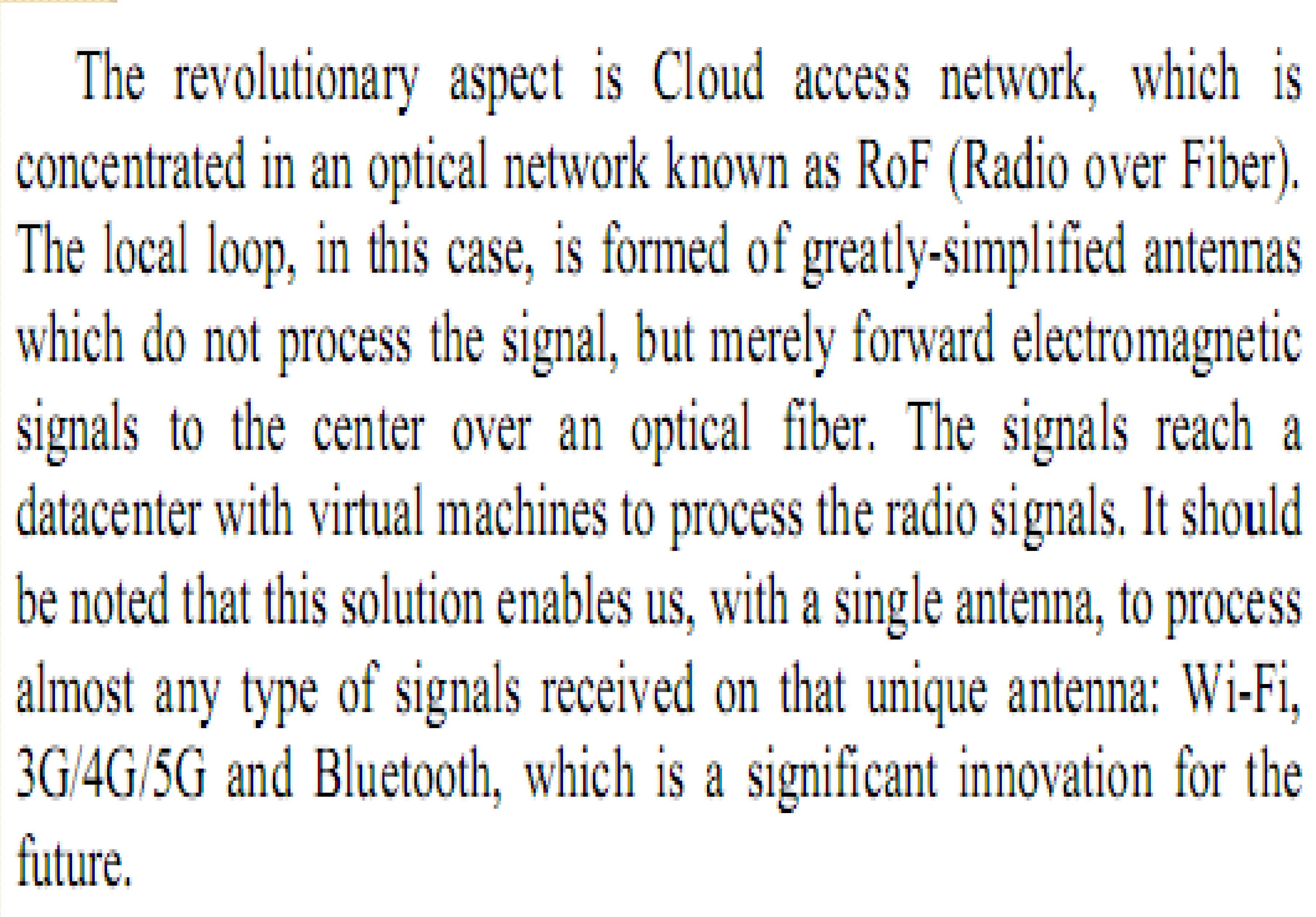
Scenarios for the placement of a controller

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



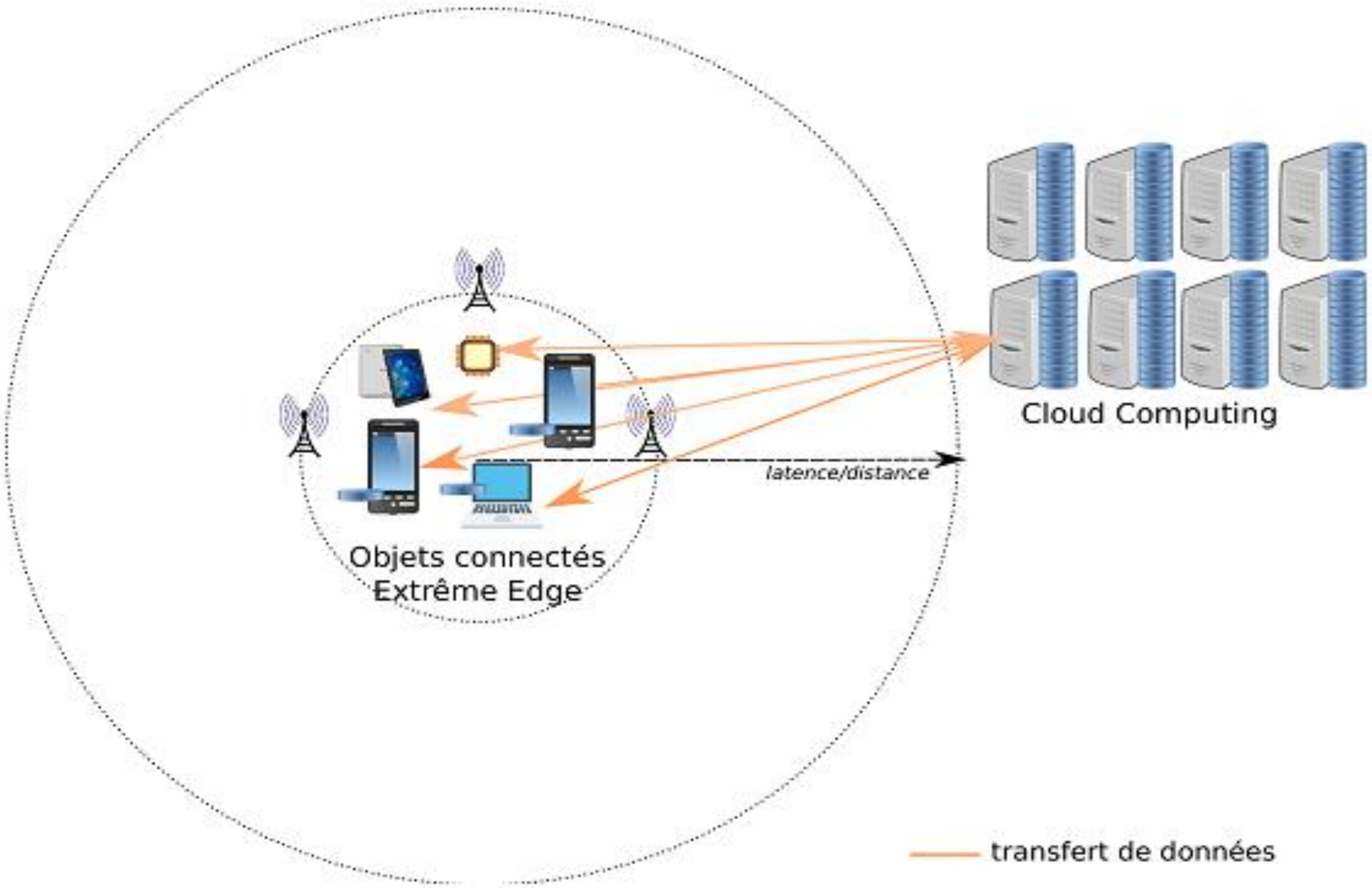
C-RAN (Cloud–Radio Access Network) / RoF (Radio over Fiber)

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



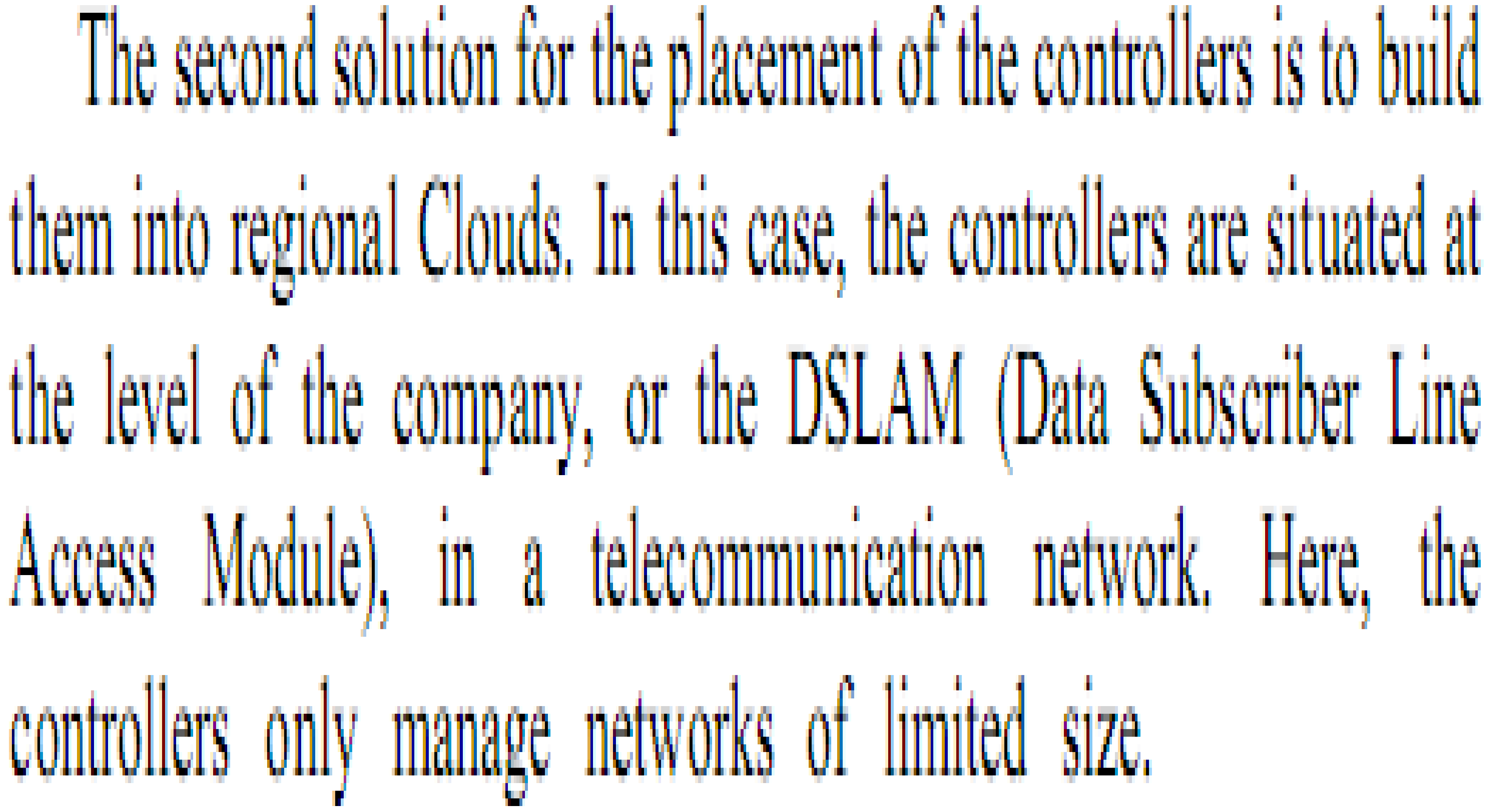
The revolutionary aspect is Cloud access network, which is concentrated in an optical network known as RoF (Radio over Fiber). The local loop, in this case, is formed of greatly-simplified antennas which do not process the signal, but merely forward electromagnetic signals to the center over an optical fiber. The signals reach a datacenter with virtual machines to process the radio signals. It should be noted that this solution enables us, with a single antenna, to process almost any type of signals received on that unique antenna: Wi-Fi, 3G/4G/5G and Bluetooth, which is a significant innovation for the future.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



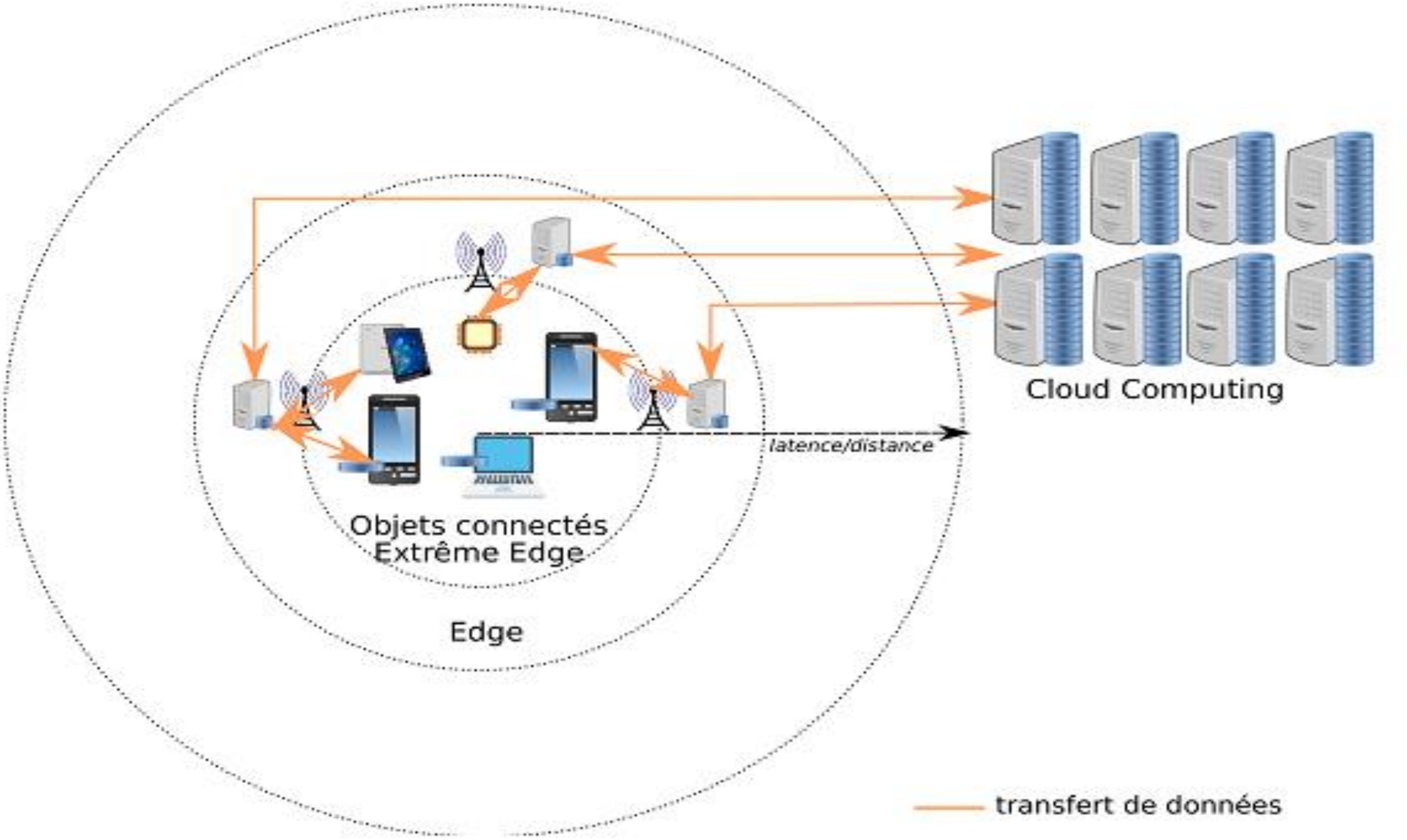
Approche Cloud Computing

Bastien Confais, Conception d'un système de partage de données adapté à un environnement de Fog Computing, 2018



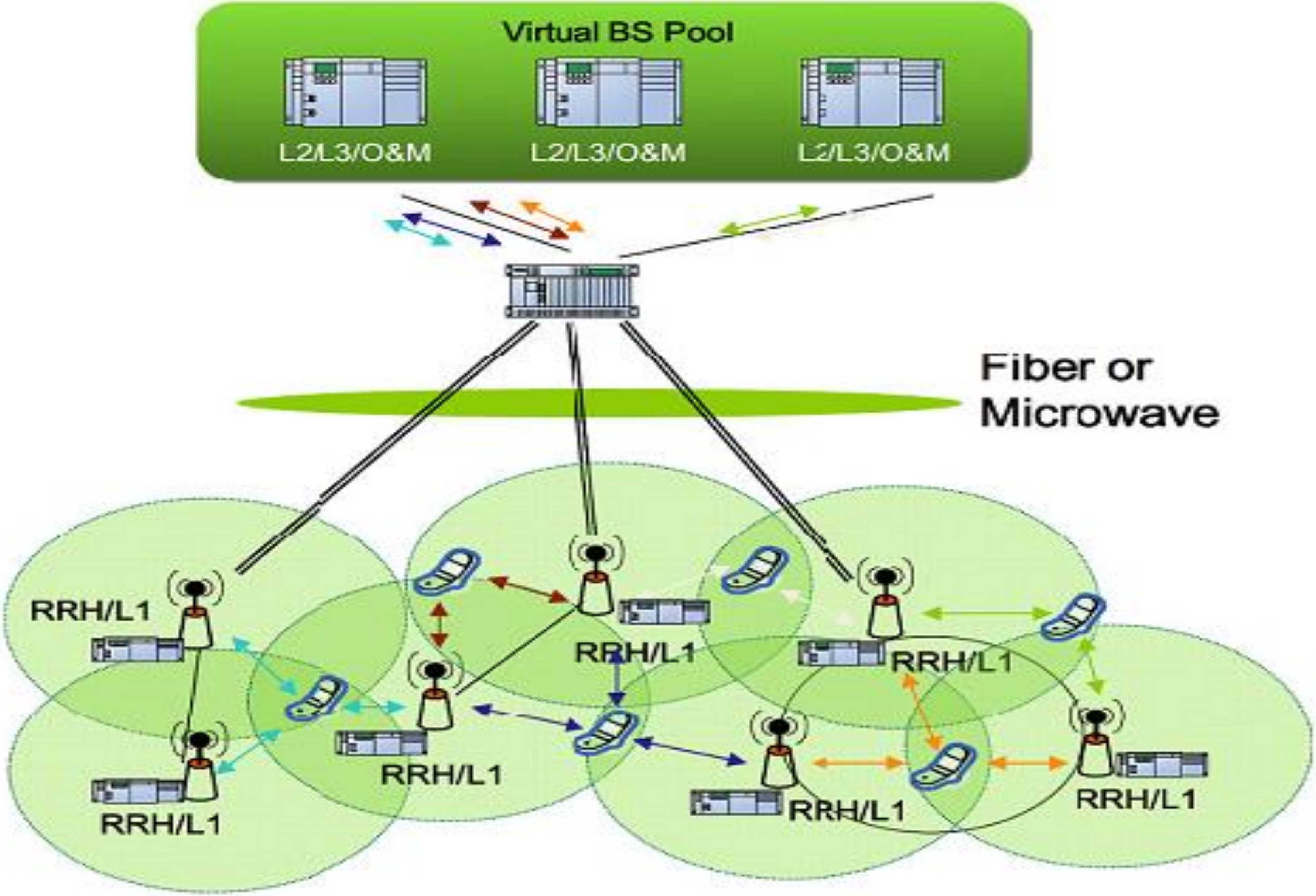
The second solution for the placement of the controllers is to build them into regional Clouds. In this case, the controllers are situated at the level of the company, or the DSLAM (Data Subscriber Line Access Module), in a telecommunication network. Here, the controllers only manage networks of limited size.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



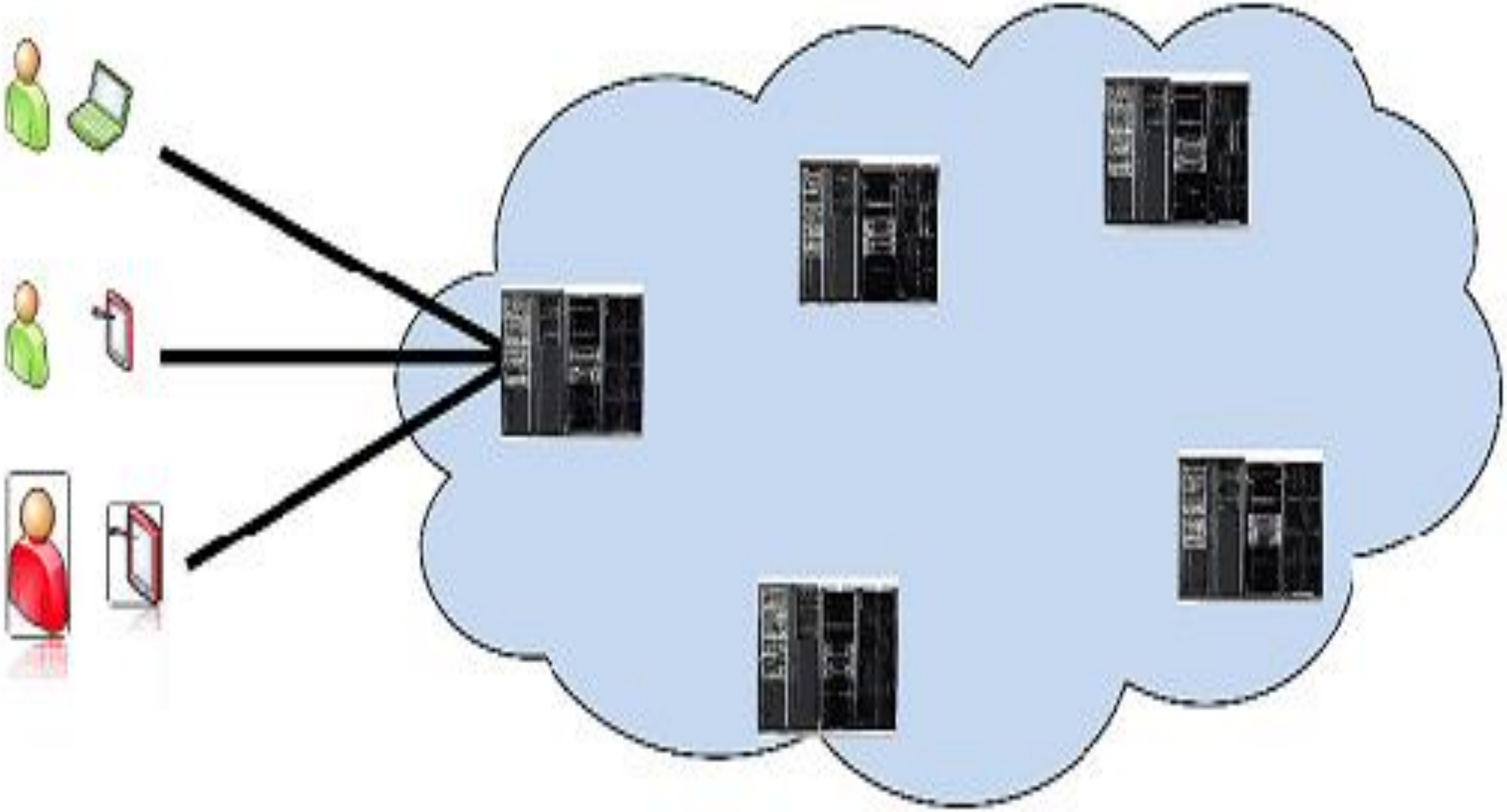
Approche Edge Computing

Bastien Confais, Conception d'un système de partage de données adapté à un environnement de Fog Computing, 2018



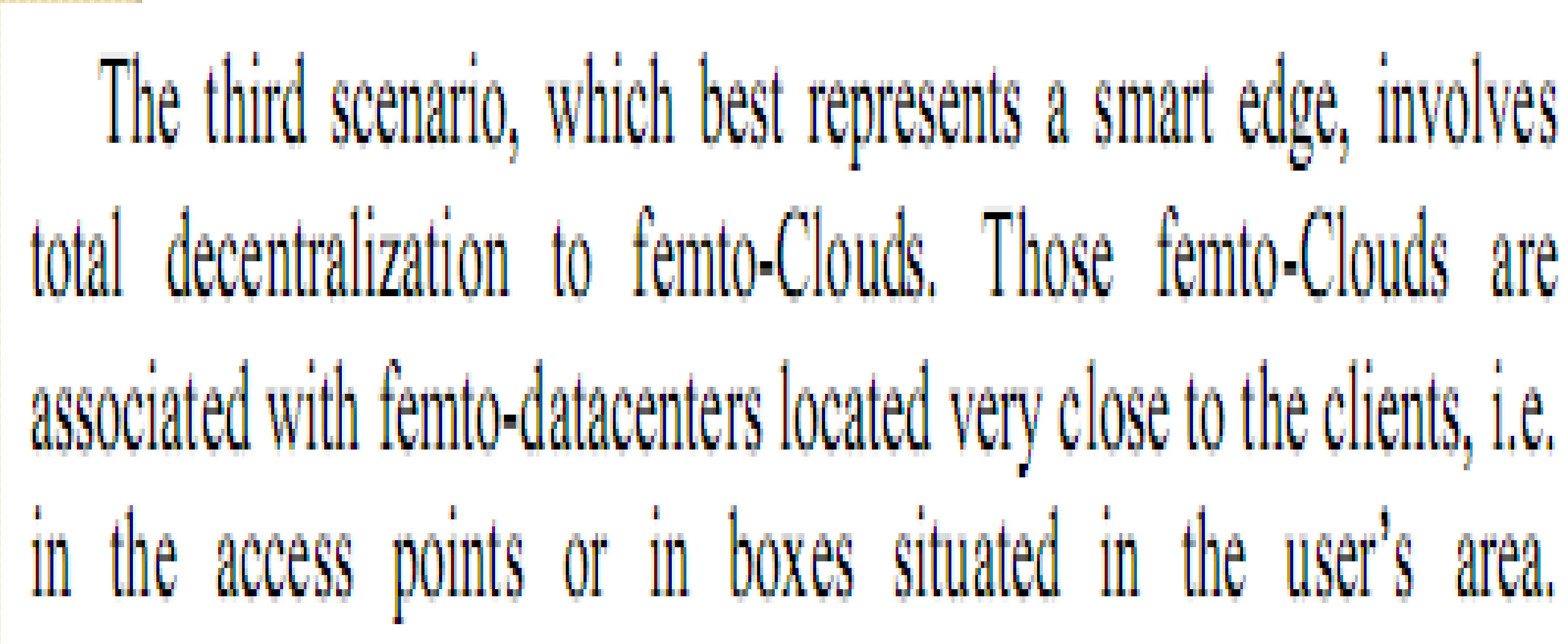
C-RAN with a distribution element

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

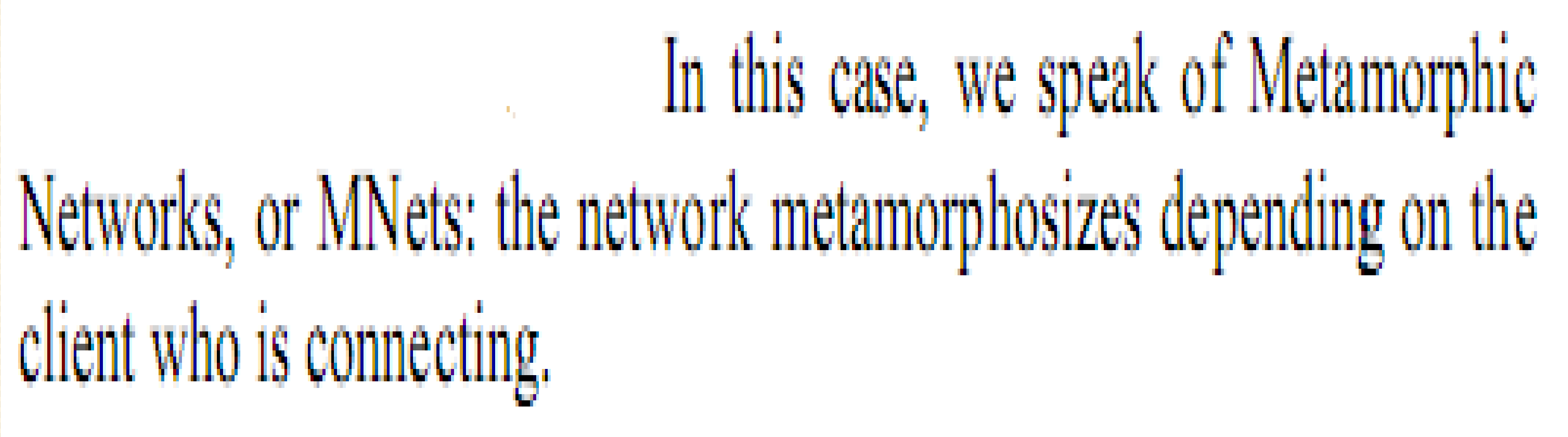


Cloudlet solution

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



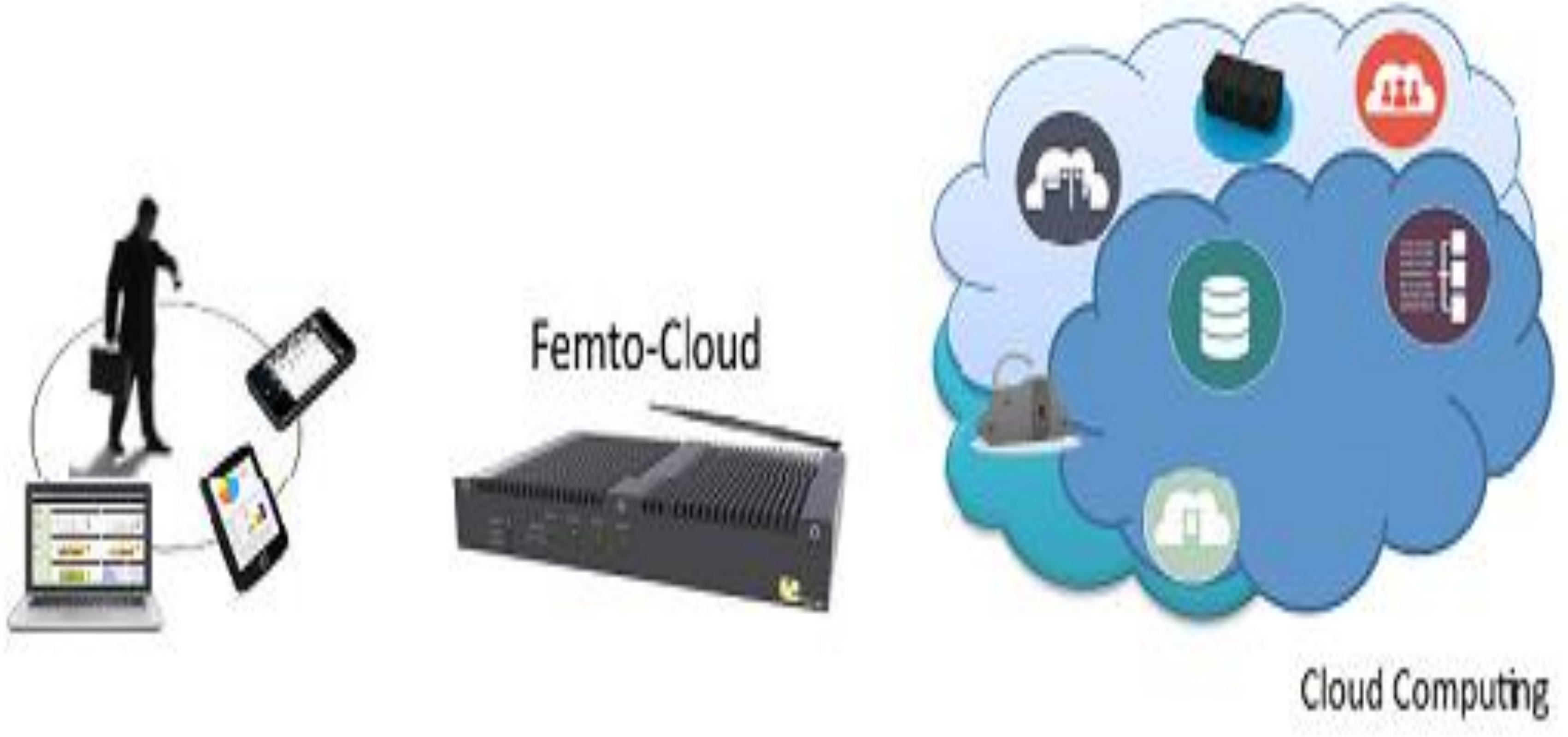
The third scenario, which best represents a smart edge, involves total decentralization to femto-Clouds. Those femto-Clouds are associated with femto-datacenters located very close to the clients, i.e. in the access points or in boxes situated in the user's area.



In this case, we speak of Metamorphic Networks, or MNets: the network metamorphosizes depending on the client who is connecting.

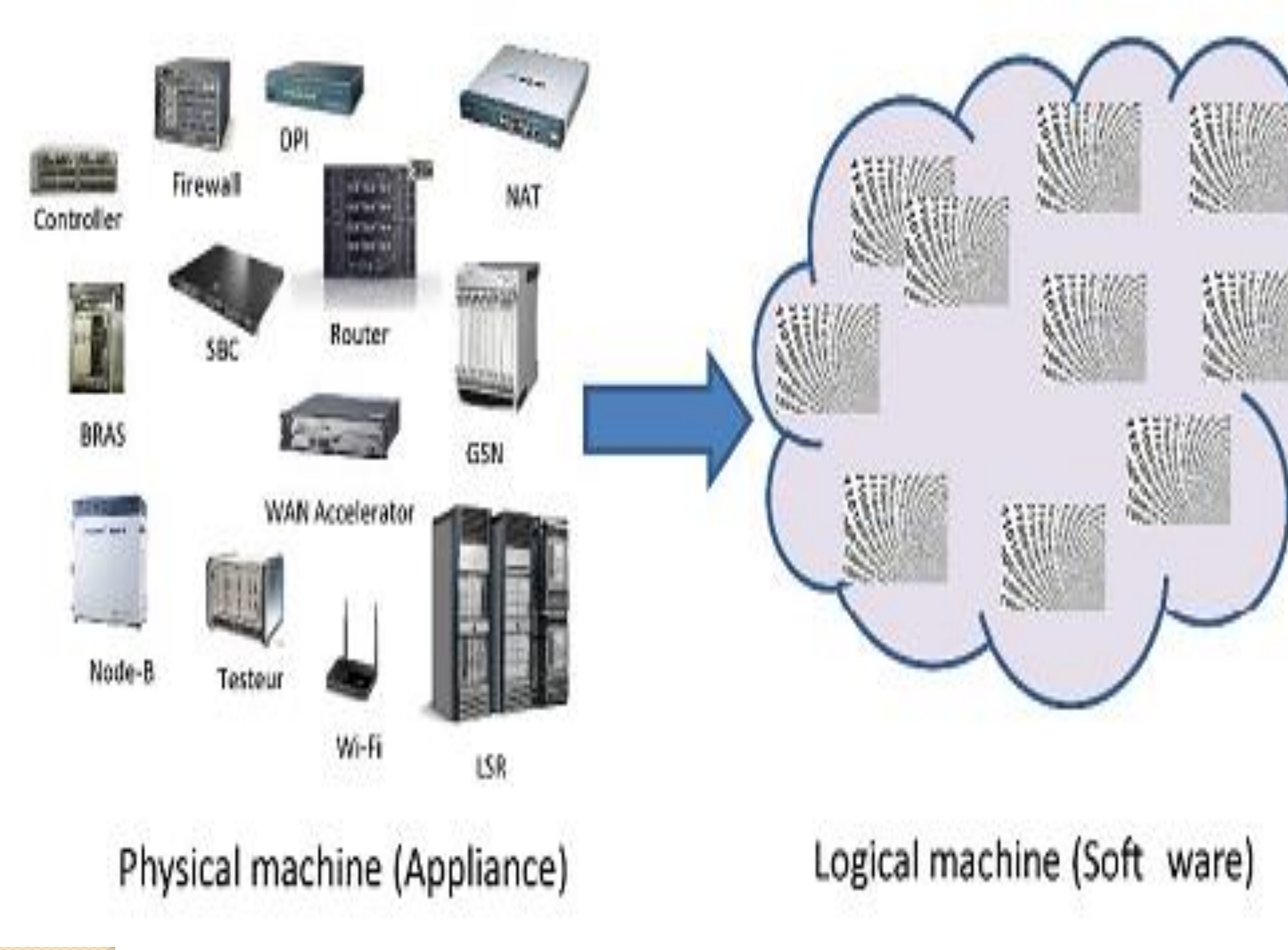
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

MNetBox (Metamorphic Network Box) from VirtuOR

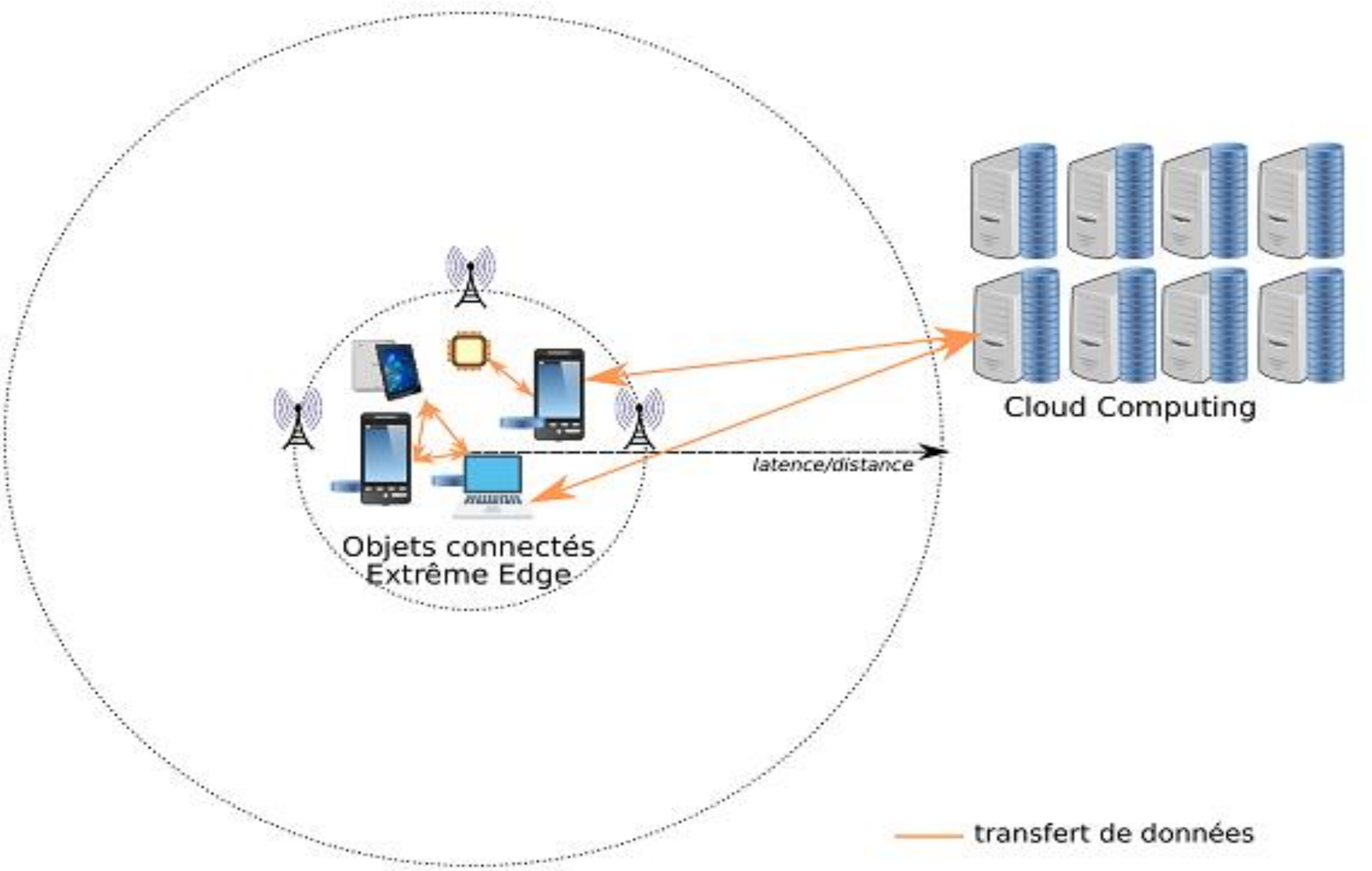


A network with a femto-datacenter

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

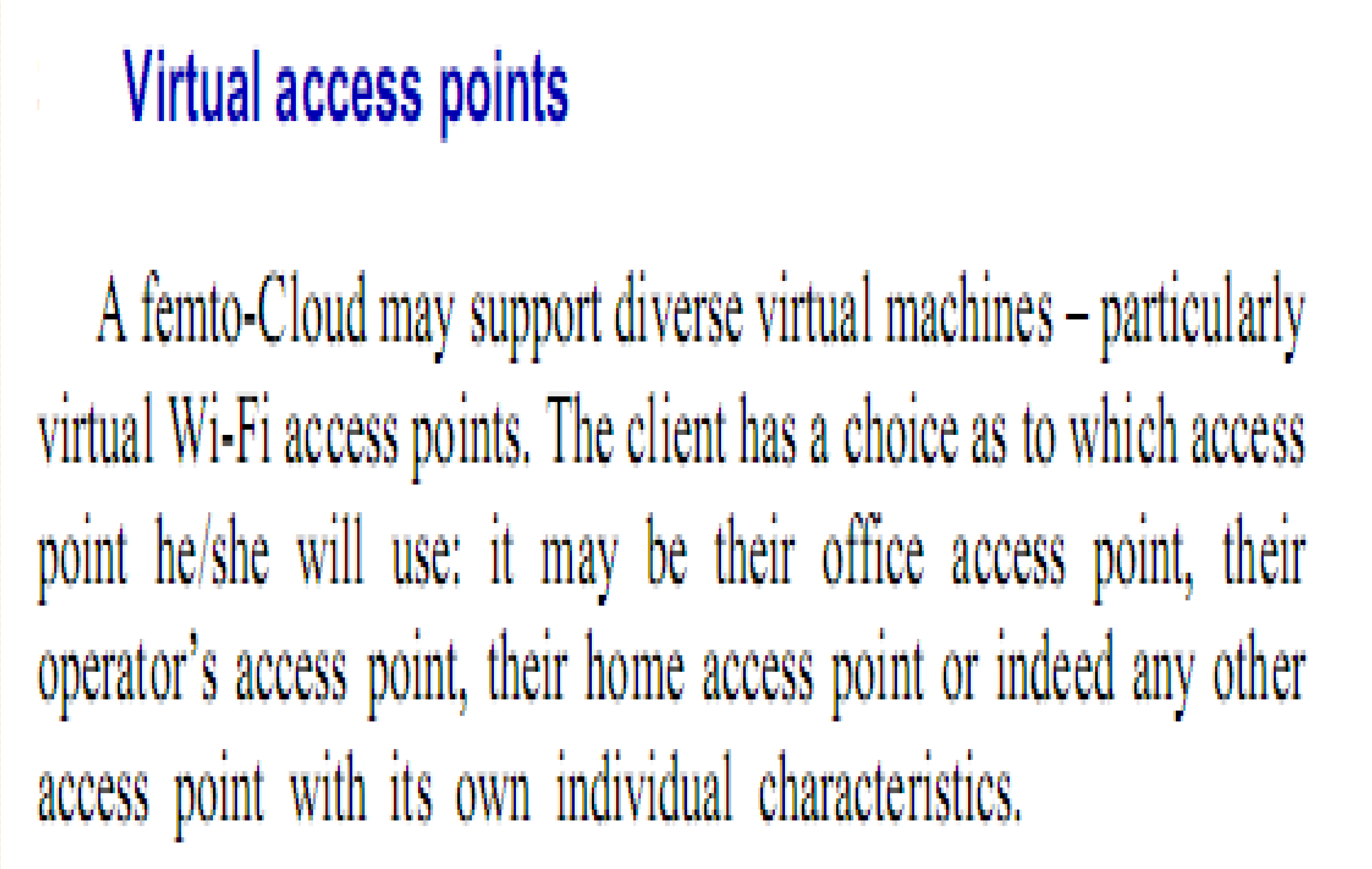


Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Approche Extrême Edge Computing

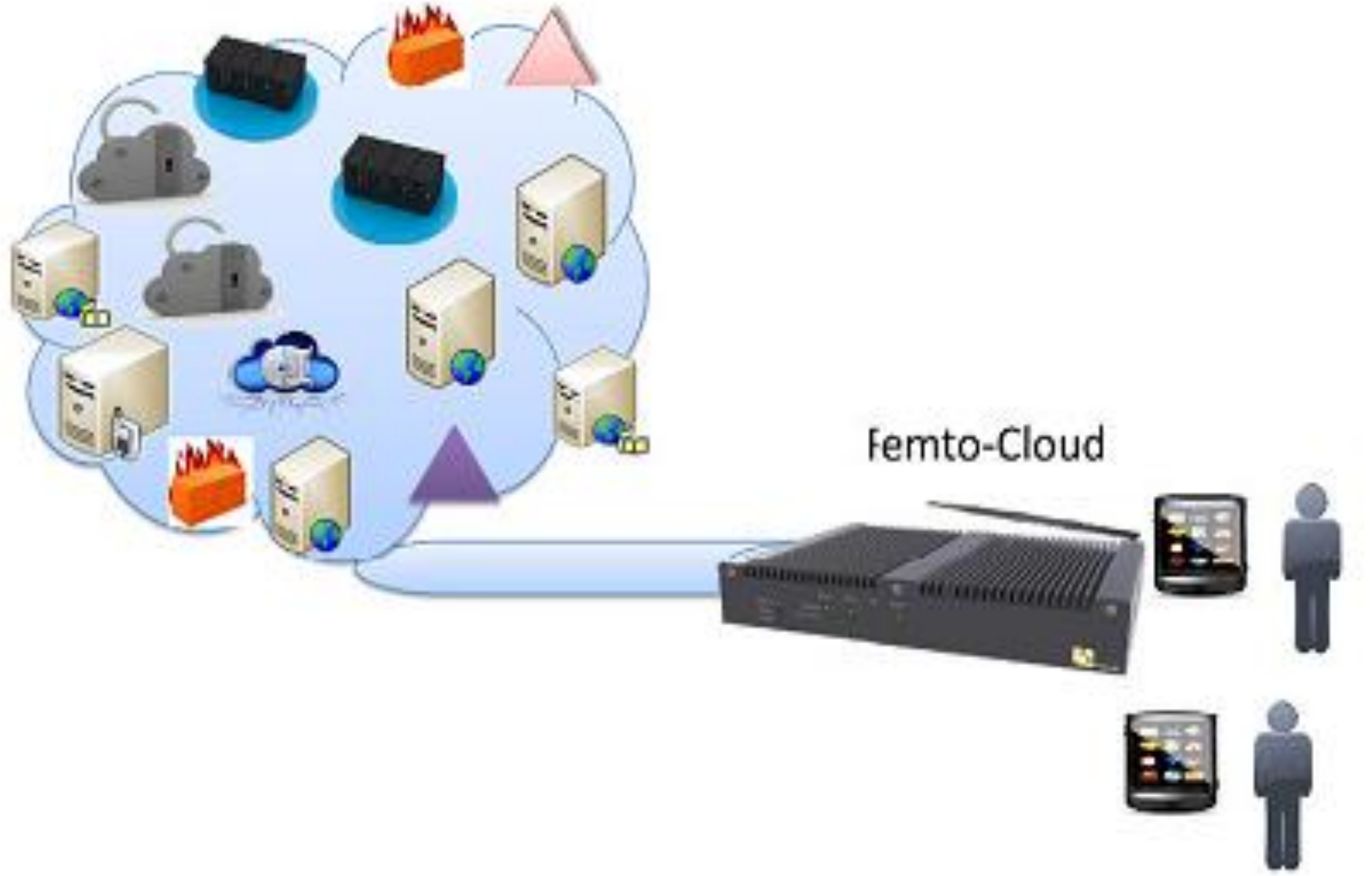
Bastien Confais, Conception d'un système de partage de données adapté à un environnement de Fog Computing, 2018



Virtual access points

A femto-Cloud may support diverse virtual machines – particularly virtual Wi-Fi access points. The client has a choice as to which access point he/she will use: it may be their office access point, their operator's access point, their home access point or indeed any other access point with its own individual characteristics.

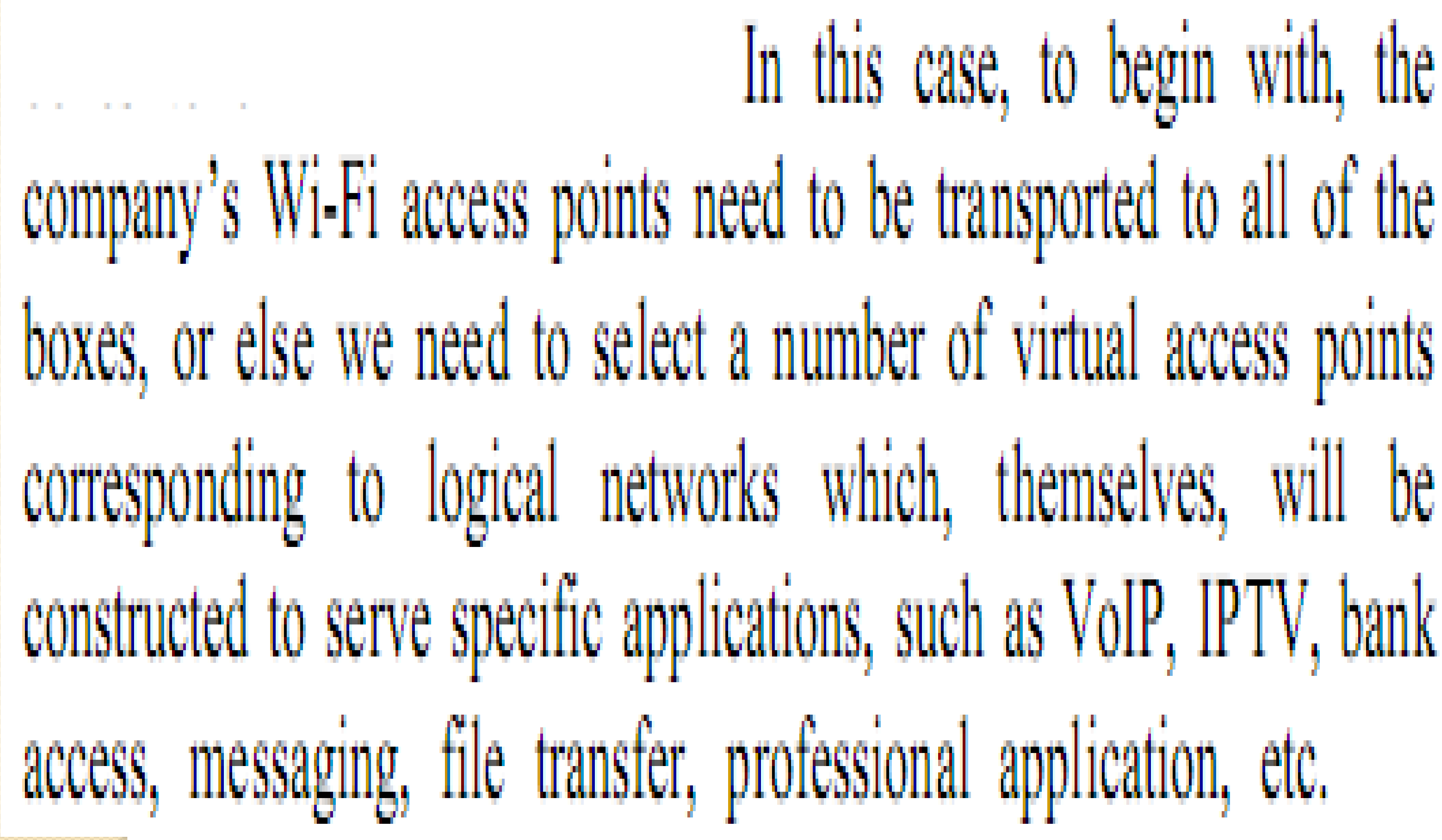
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Context of a femto-Cloud network for a "smart edge"

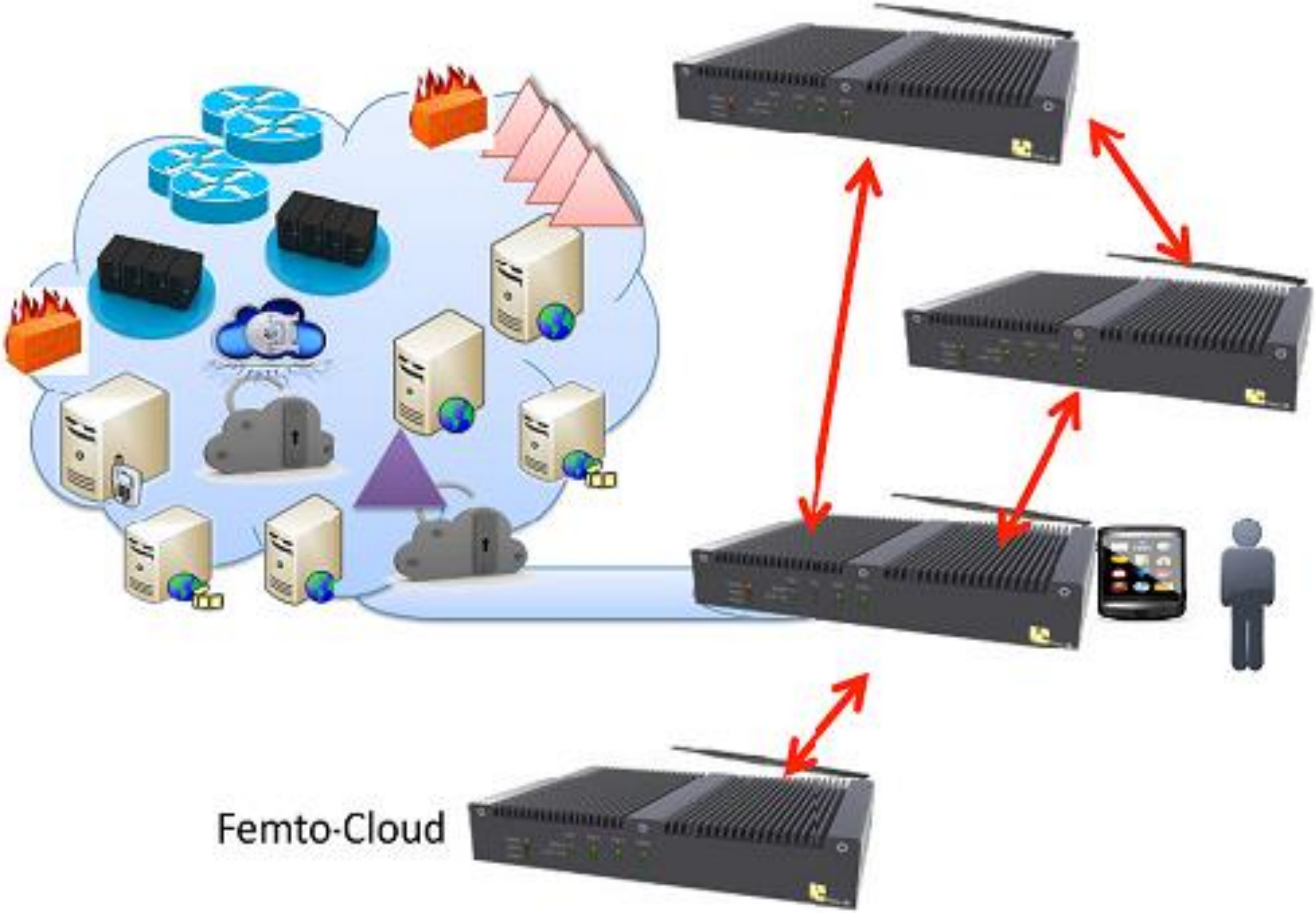
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

Software LANs



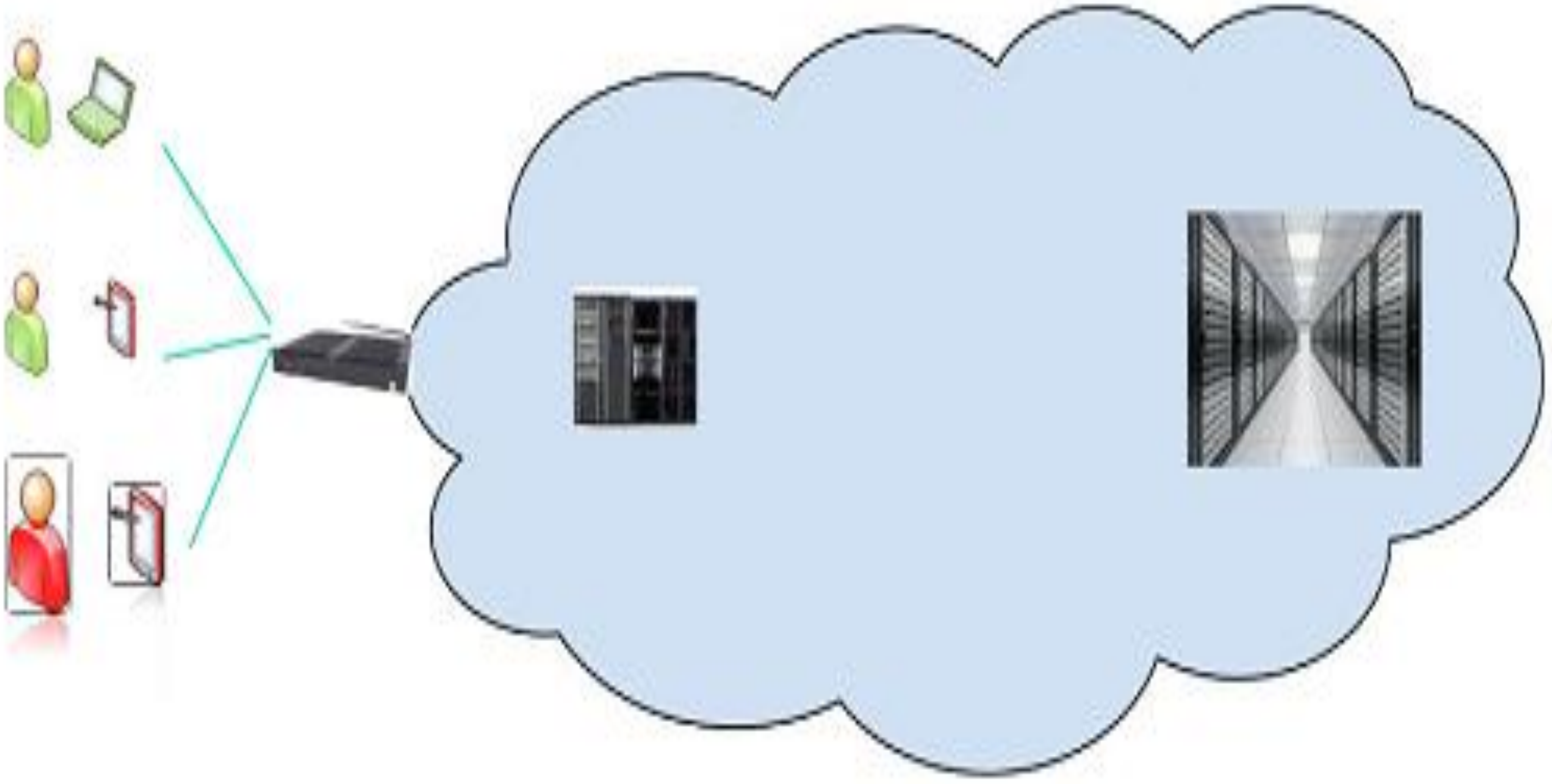
In this case, to begin with, the company's Wi-Fi access points need to be transported to all of the boxes, or else we need to select a number of virtual access points corresponding to logical networks which, themselves, will be constructed to serve specific applications, such as VoIP, IPTV, bank access, messaging, file transfer, professional application, etc.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



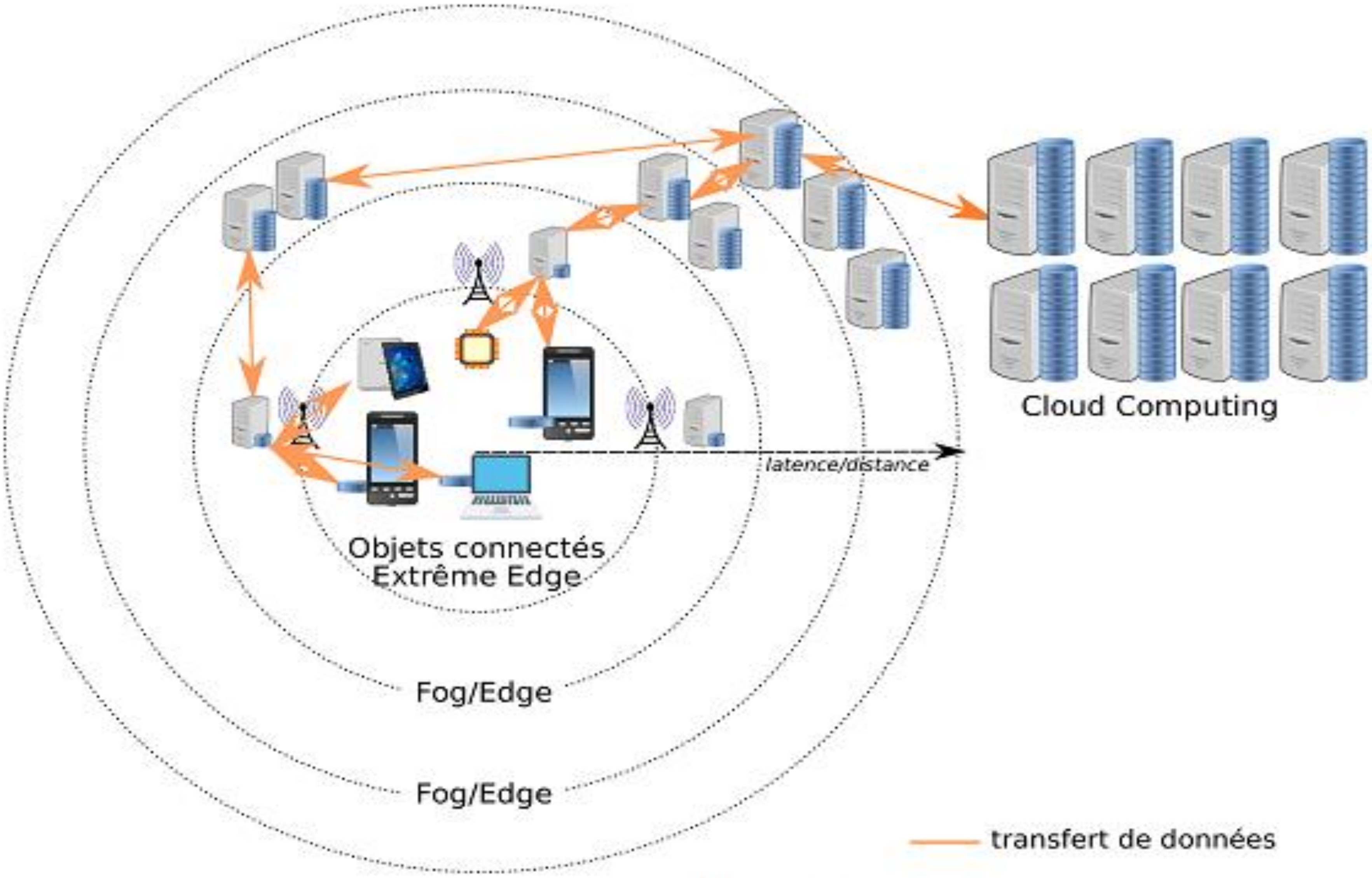
A femto-datacenter environment to create virtual LANs

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



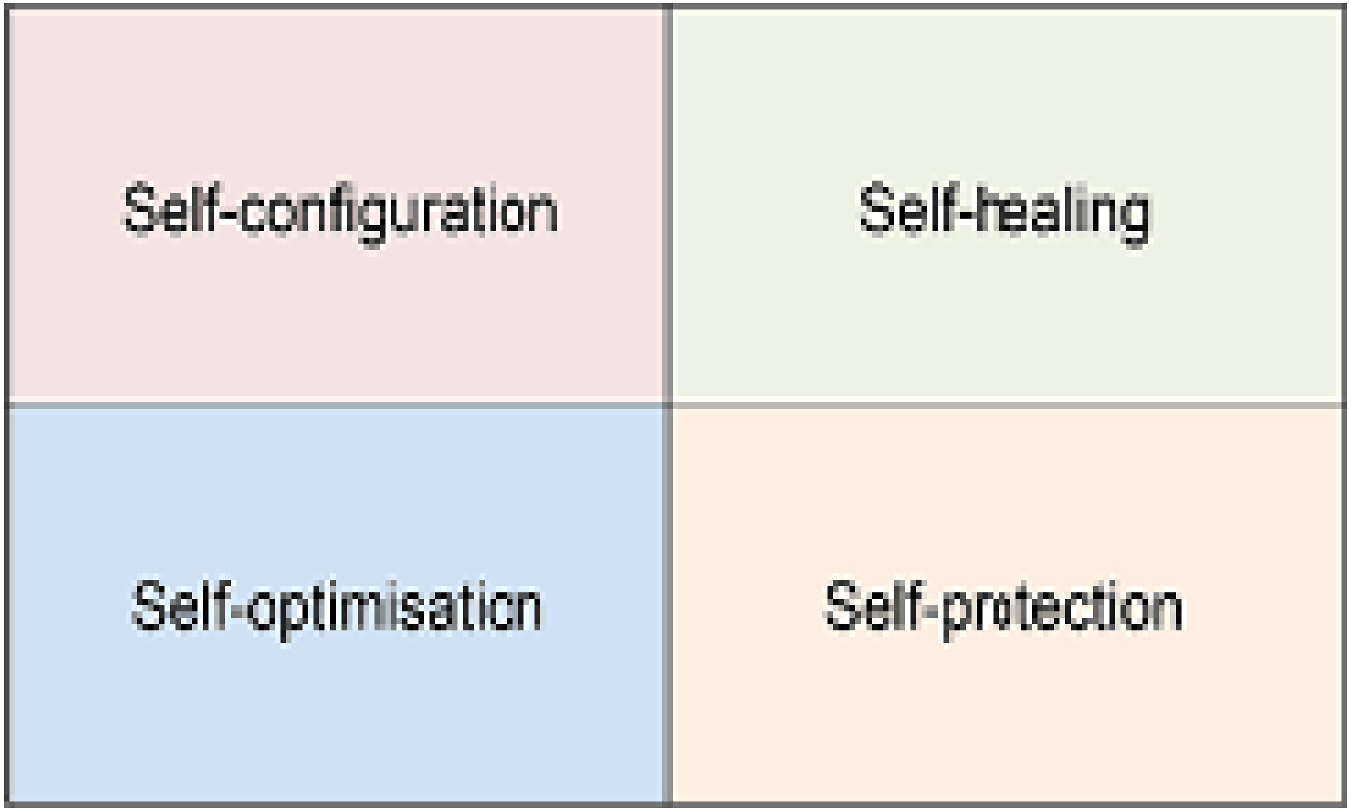
Hierarchy of controls and datacenters

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Approche Fog Computing

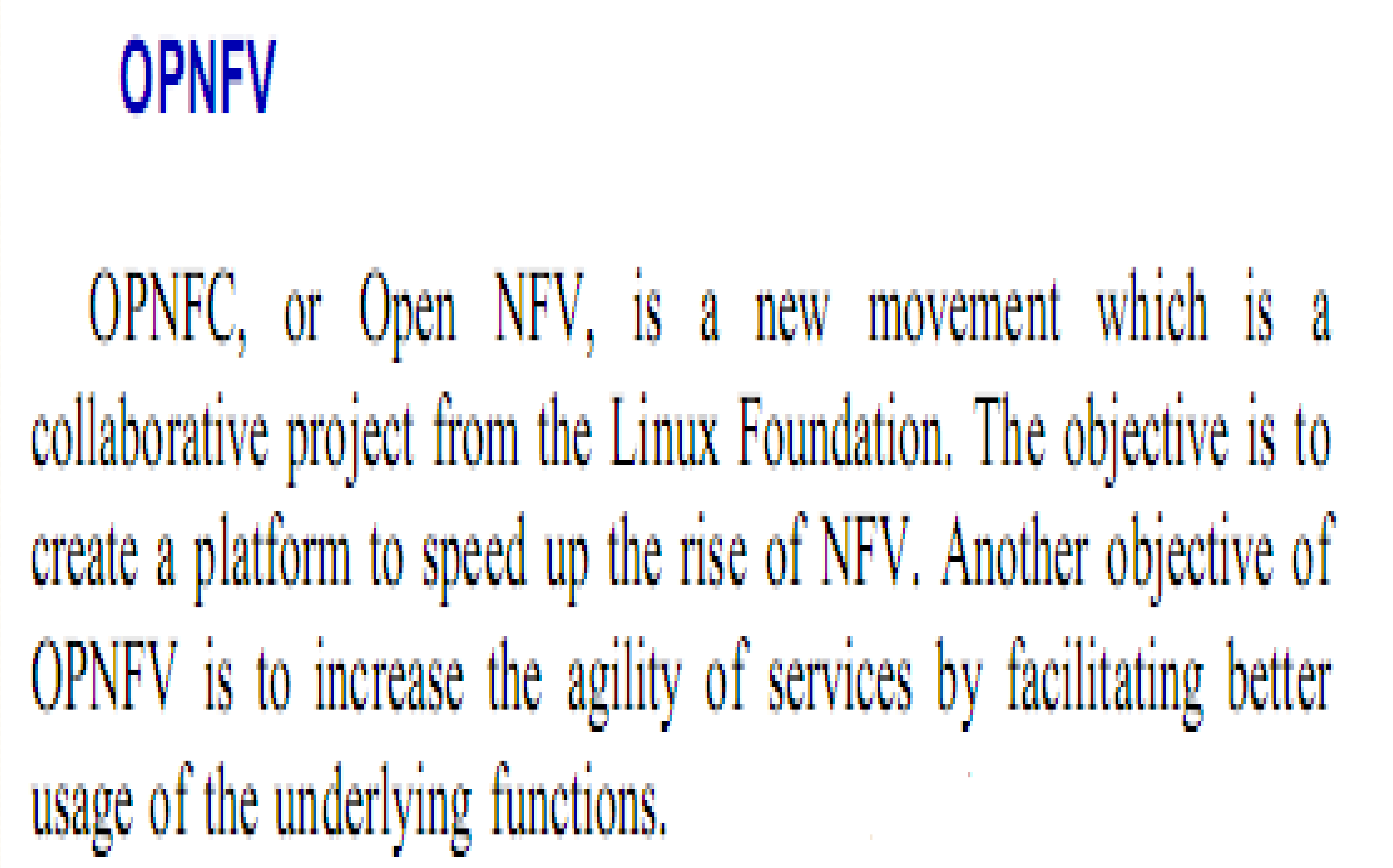
Bastien Confais, Conception d'un système de partage de données adapté à un environnement de Fog Computing, 2018



Self-configuration	Self-healing
Self-optimisation	Self-protection

Definition of an autonomic network

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



OPNFV

OPNFC, or Open NFV, is a new movement which is a collaborative project from the Linux Foundation. The objective is to create a platform to speed up the rise of NFV. Another objective of OPNFV is to increase the agility of services by facilitating better usage of the underlying functions.

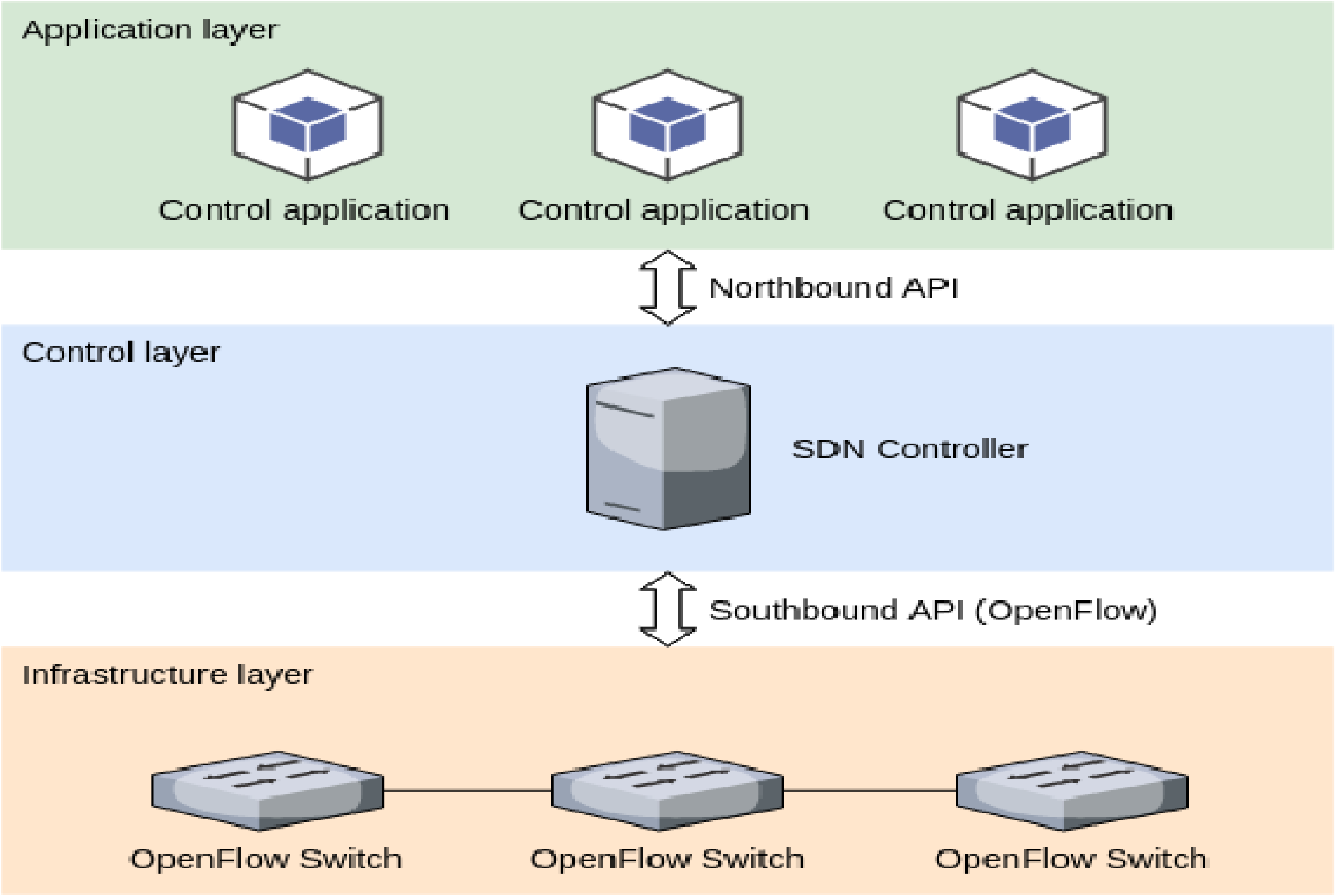
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

OpenFlow defines a protocol for communication between a centralized OpenFlow controller and OpenFlow enabled switches.

The OpenFlow channel is run over TCP.

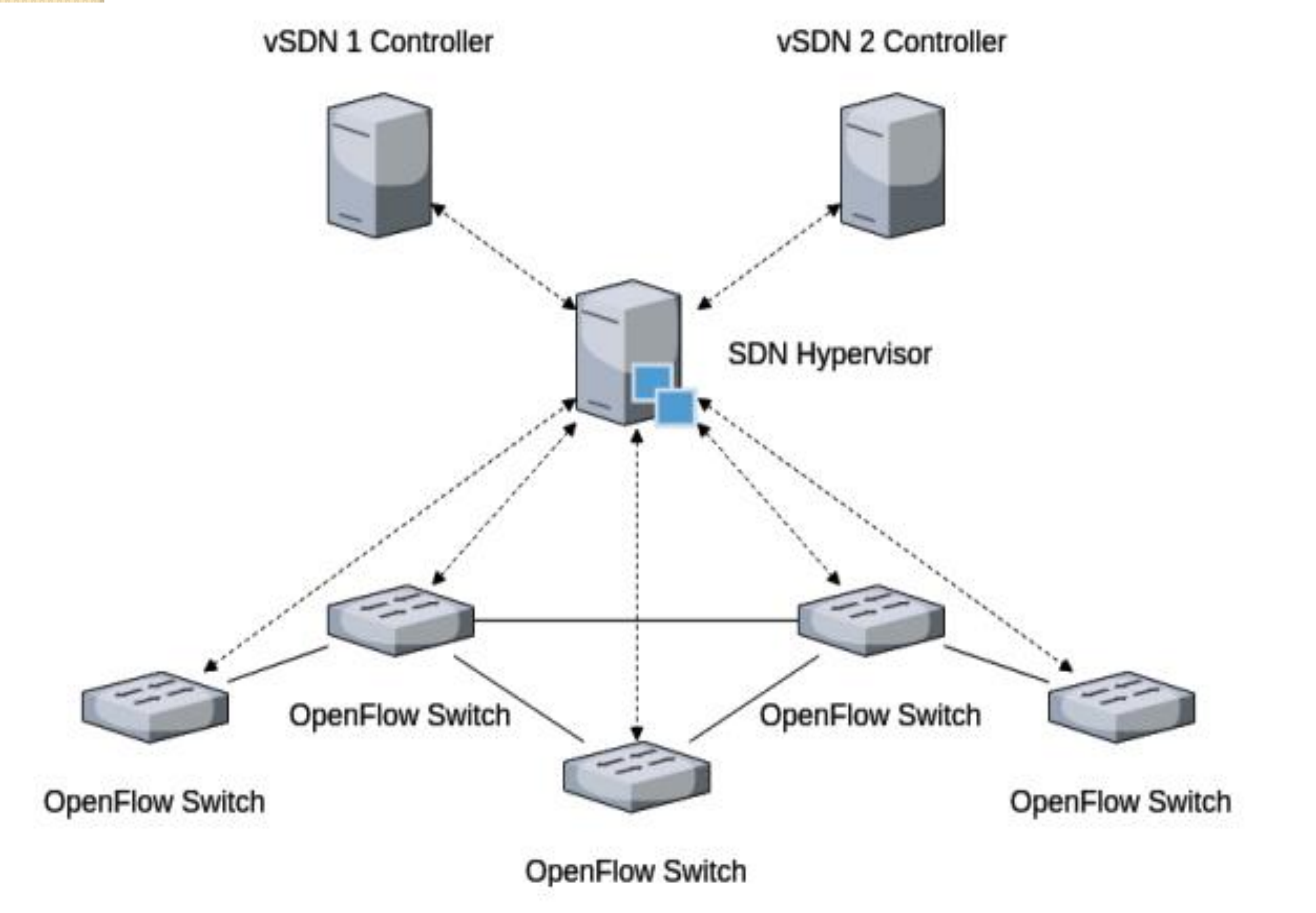
The connection is usually encrypted using TLS for security.

Source : Thomas Langenskiöld, Network Slicing using Switch Virtualization, 2017



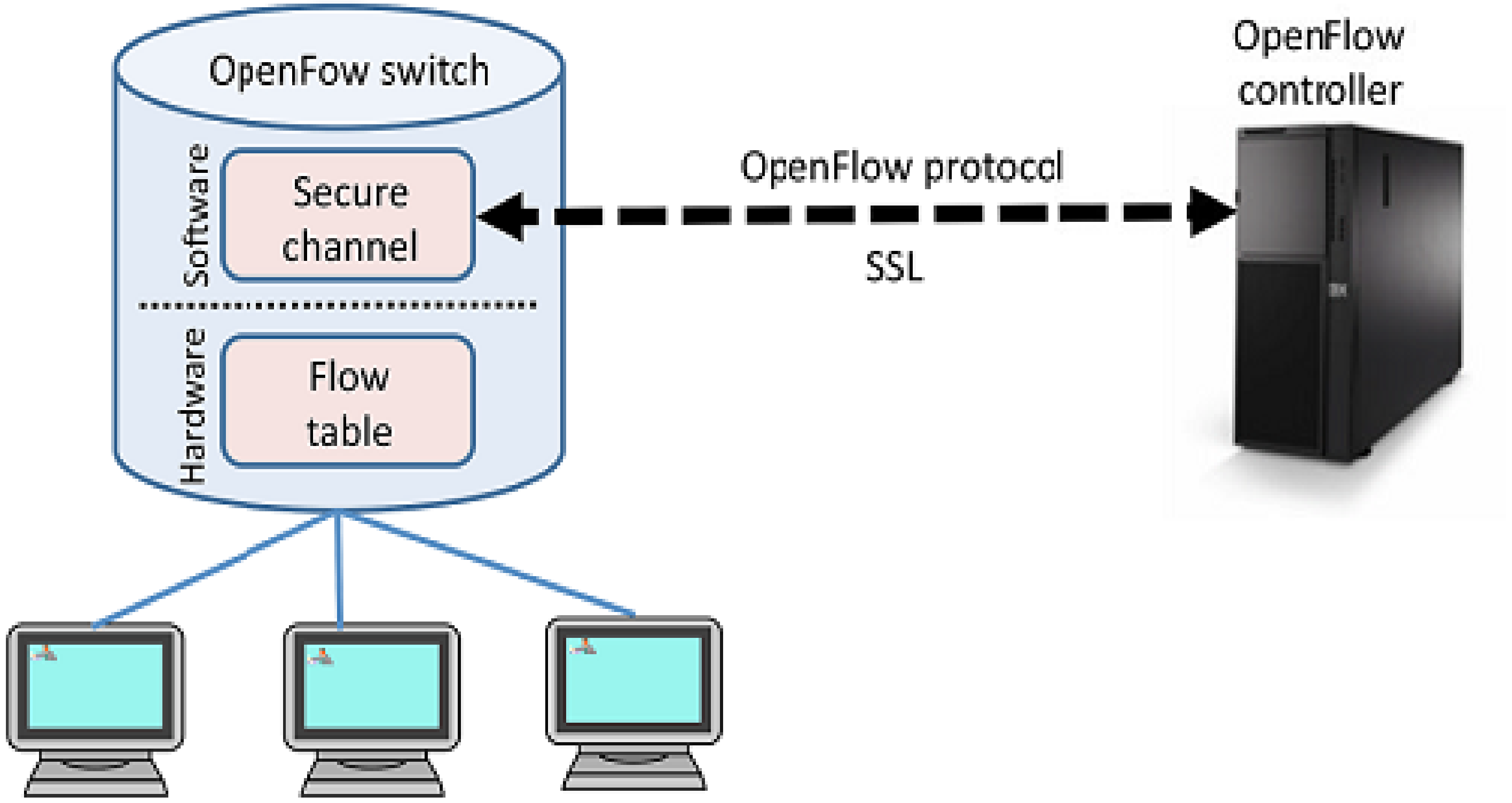
The OpenFlow SDN architecture

Source : Thomas Langenskiöld, Network Slicing using Switch Virtualization, 2017



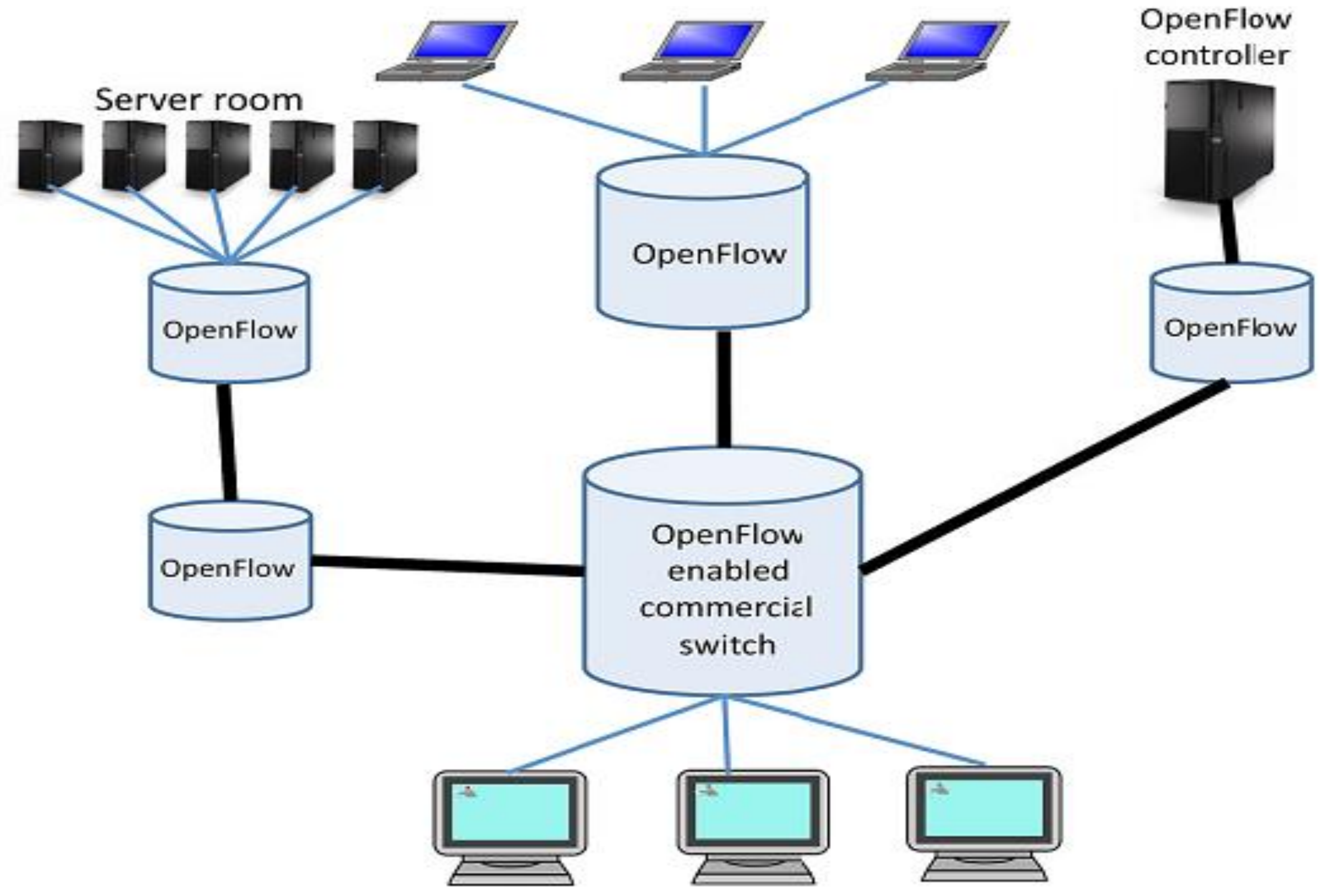
Source : Thomas Langenskiöld, Network Slicing using Switch Virtualization, 2017

OpenFlow



OpenFlow protocol

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



OpenFlow protocol in a network

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

OpFlex

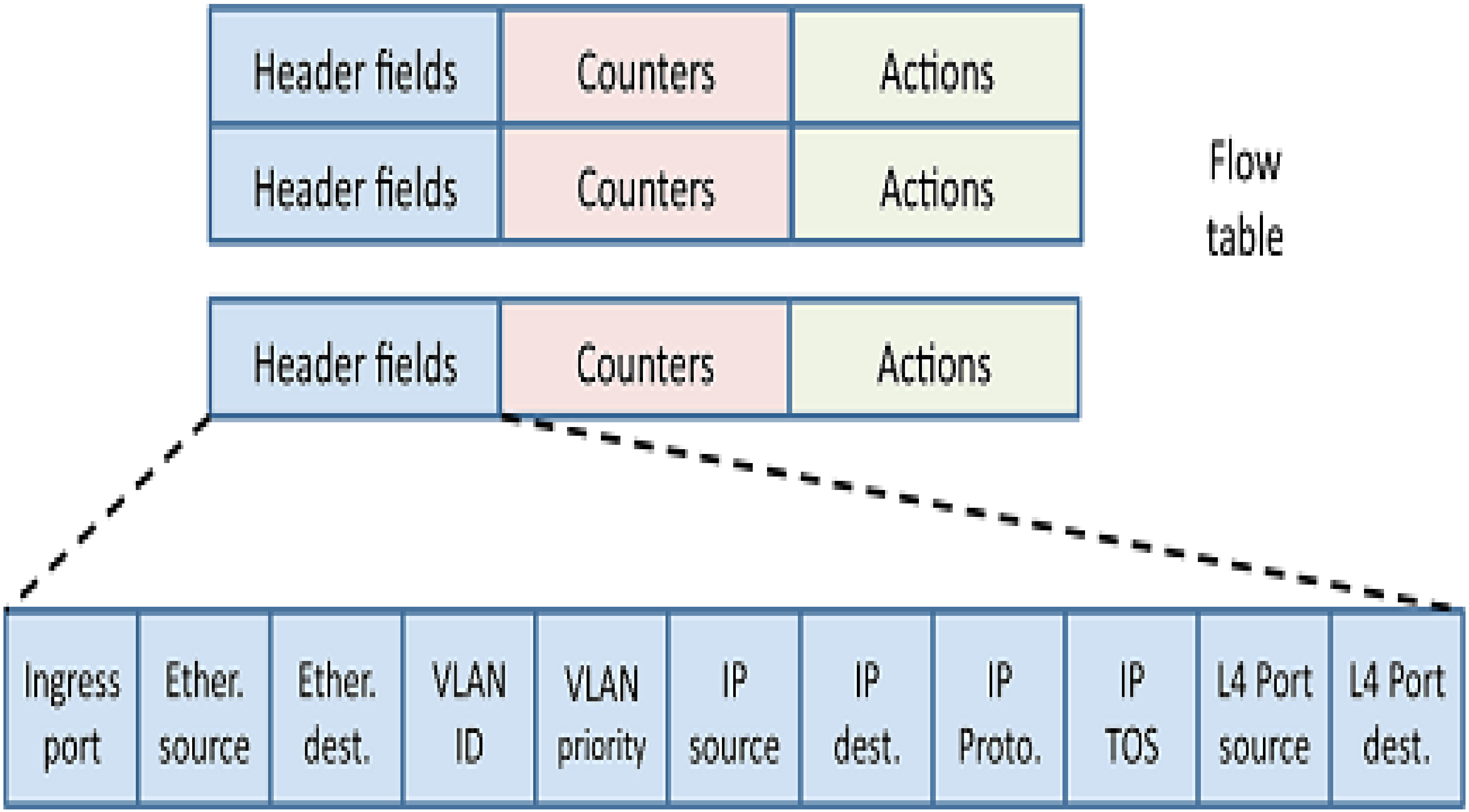
Shortly after the emergence of OpenFlow, Cisco announced an alternative southbound API, OpFlex.

OpFlex is similar to OpenFlow, but instead of controlling the behavior of the network through software applications, in OpFlex the network is managed by controlling policies. These policies are communicated to the network elements, that enforce them using traditional distributed network protocols.

Cisco believes that by moving some of the control logic back into the network elements, the risk for the controller becoming a bottleneck for network efficiency is diminished.

OpFlex needs to have an agent installed on the network elements in order to work, which has led to a lack of support for the OpFlex protocol by different vendors.

Source : Thomas Langenskiöld, Network Slicing using Switch Virtualization, 2017



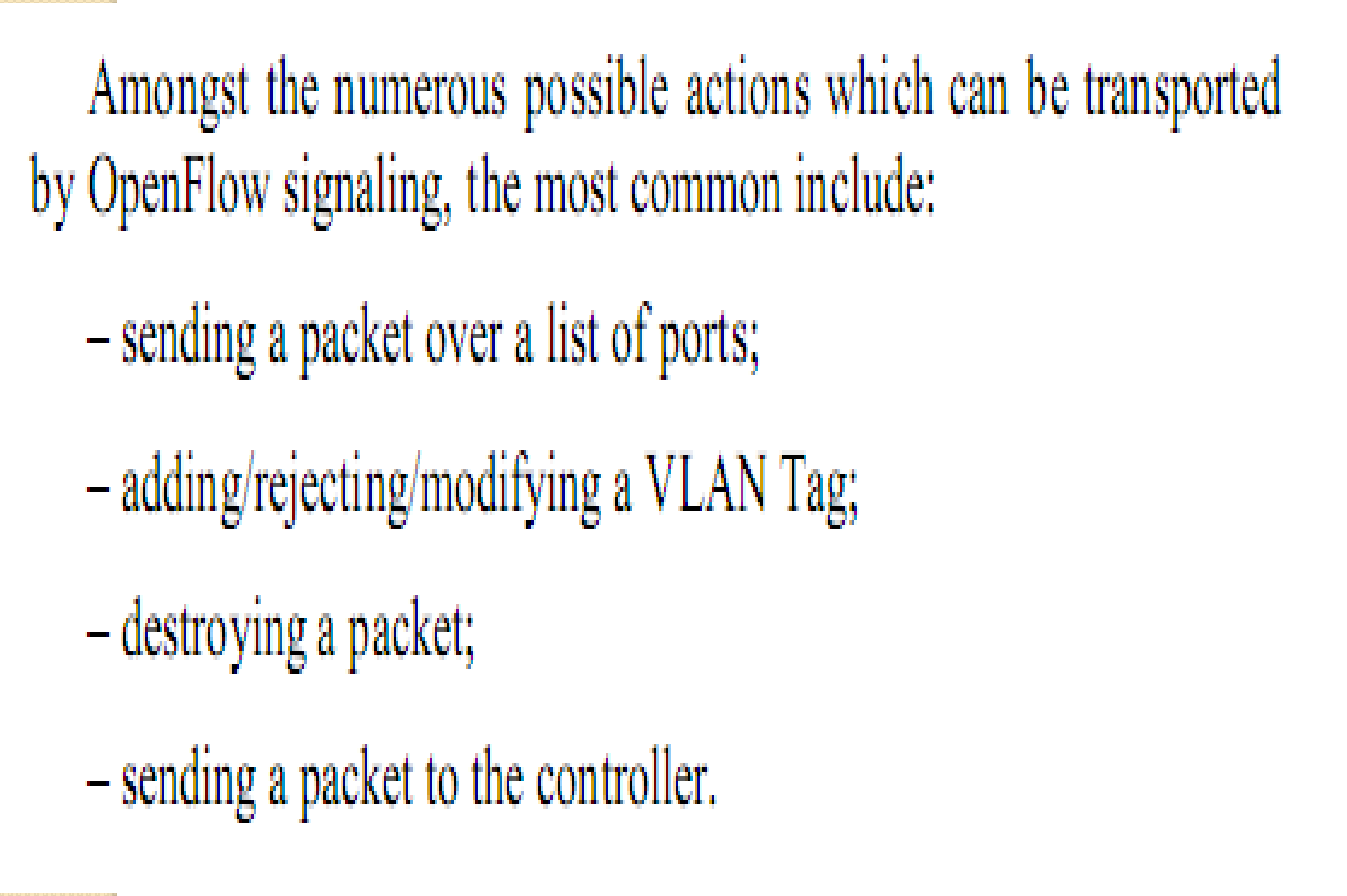
Flow table

Header fields	Counters	Actions
Header fields	Counters	Actions
Header fields	Counters	Actions

Ingress port	Ether. source	Ether. dest.	VLAN ID	VLAN priority	IP source	IP dest.	IP Proto.	IP TOS	L4 Port source	L4 Port dest.

Fields in the OpenFlow protocol

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Amongst the numerous possible actions which can be transported by OpenFlow signaling, the most common include:

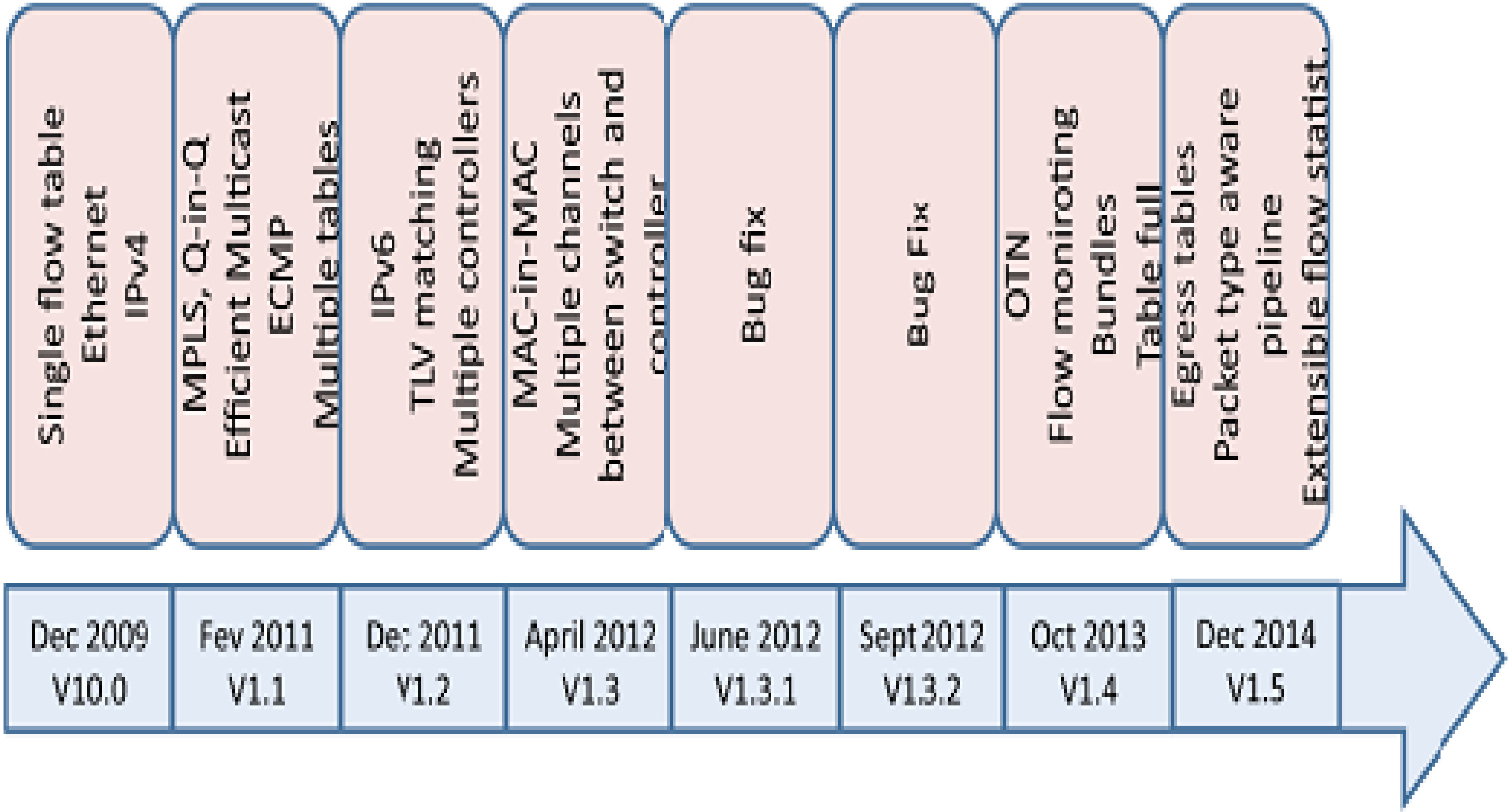
– sending a packet over a list of ports;

– adding/rejecting/modifying a VLAN Tag;

– destroying a packet;

– sending a packet to the controller.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Single flow table
Ethernet
IPv4

MPLS, Q-in-Q
Efficient Multicast
ECMP
Multiple tables

IPv6
TLV matching
Multiple controllers

MAC-in-MAC
Multiple channels
between switch and
controller

Bug fix

Bug Fix

OTN
Flow moniroting
Bundles
Table full

Egress tables
Packet type aware
pipeline
Extensible flow statist.

Dec 2009
V10.0

Fev 2011
V1.1

De: 2011
V1.2

April 2012
V1.3

June 2012
V1.3.1

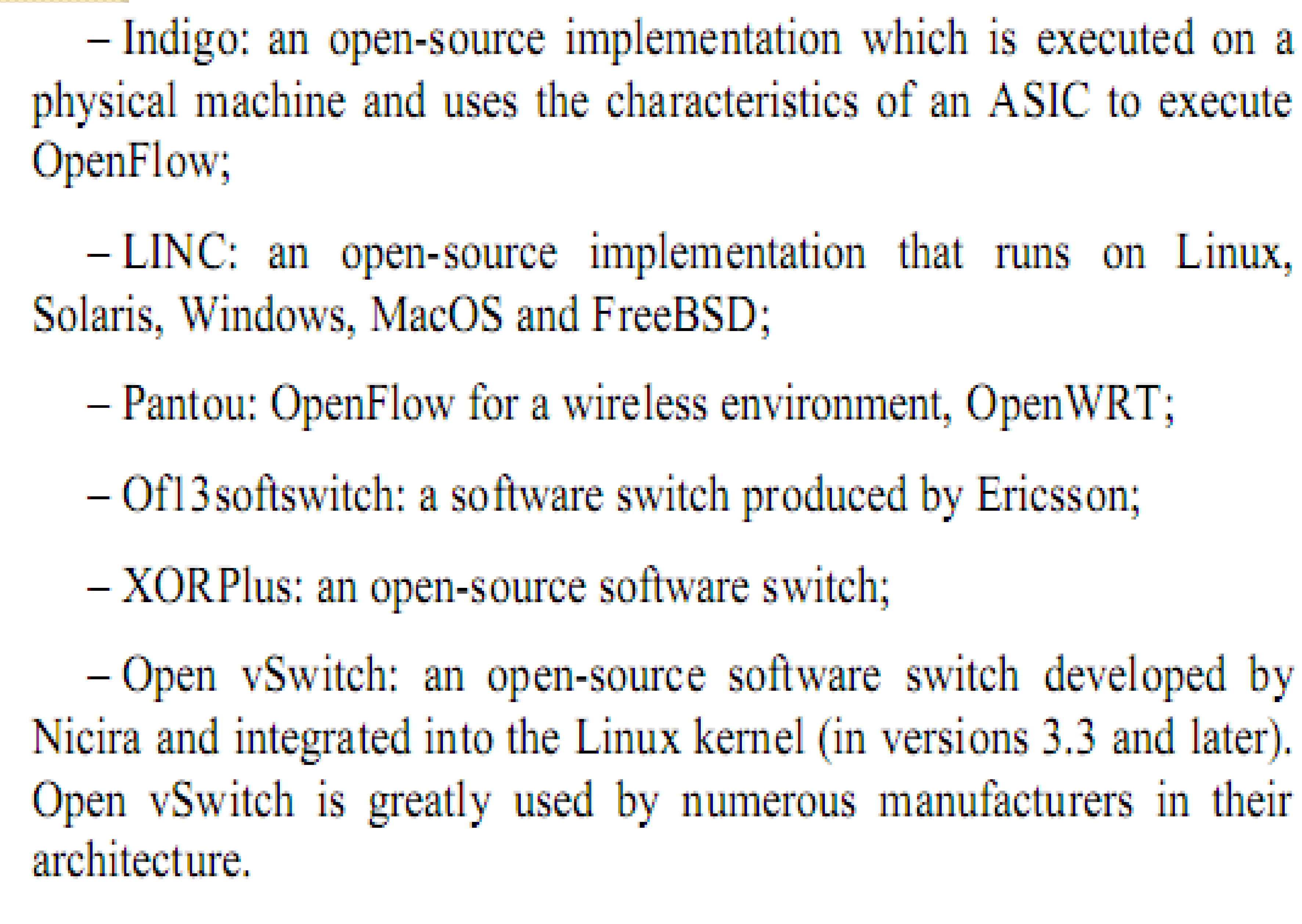
Sept 2012
V1.3.2

Oct 2013
V1.4

Dec 2014
V1.5

The different ONF standards pertaining to the OpenFlow protocol

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



– Indigo: an open-source implementation which is executed on a physical machine and uses the characteristics of an ASIC to execute OpenFlow;

– LINC: an open-source implementation that runs on Linux, Solaris, Windows, MacOS and FreeBSD;

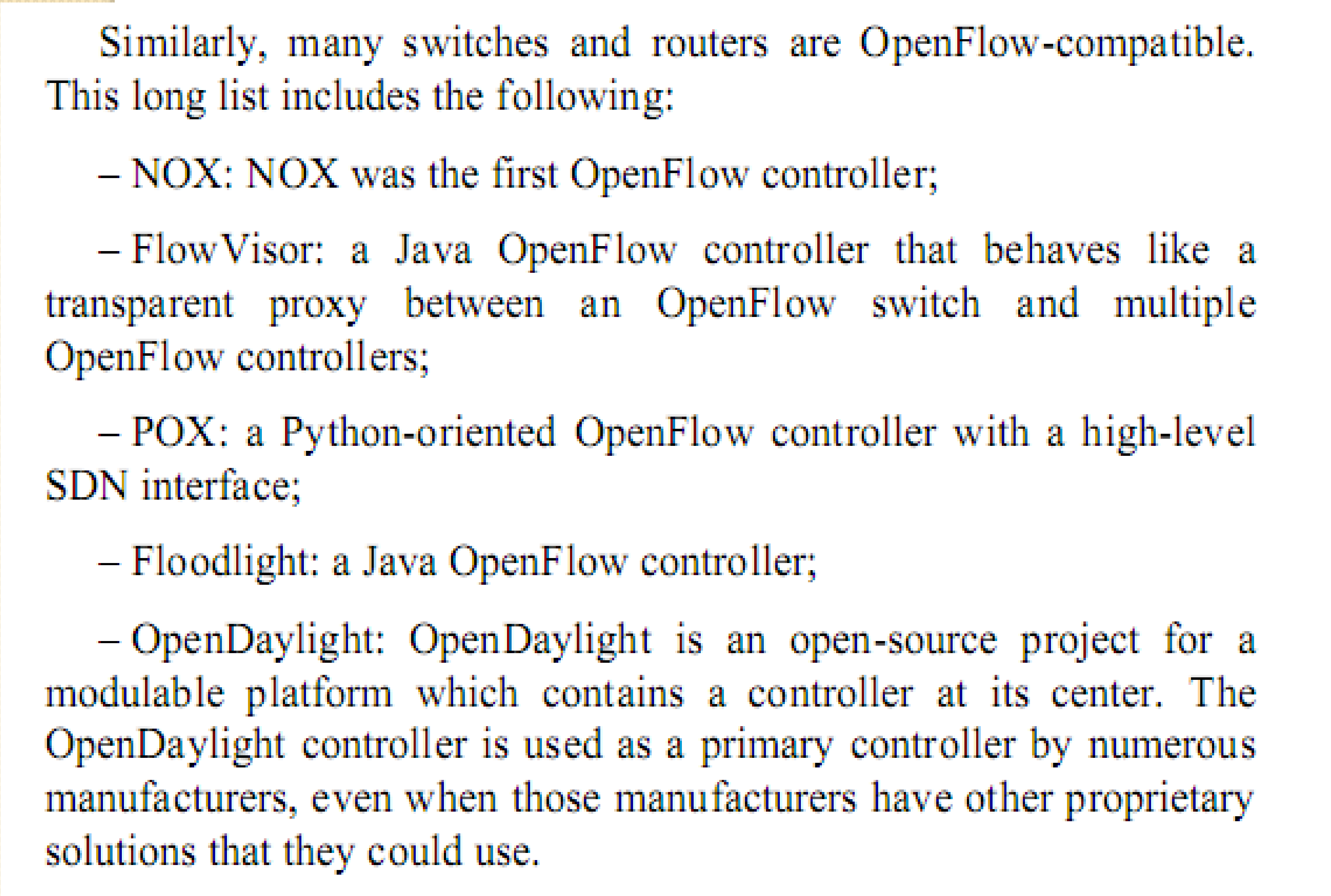
– Pantou: OpenFlow for a wireless environment, OpenWRT;

– Of13softswitch: a software switch produced by Ericsson;

– XORPlus: an open-source software switch;

– Open vSwitch: an open-source software switch developed by Nicira and integrated into the Linux kernel (in versions 3.3 and later). Open vSwitch is greatly used by numerous manufacturers in their architecture.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Similarly, many switches and routers are OpenFlow-compatible. This long list includes the following:

– NOX: NOX was the first OpenFlow controller;

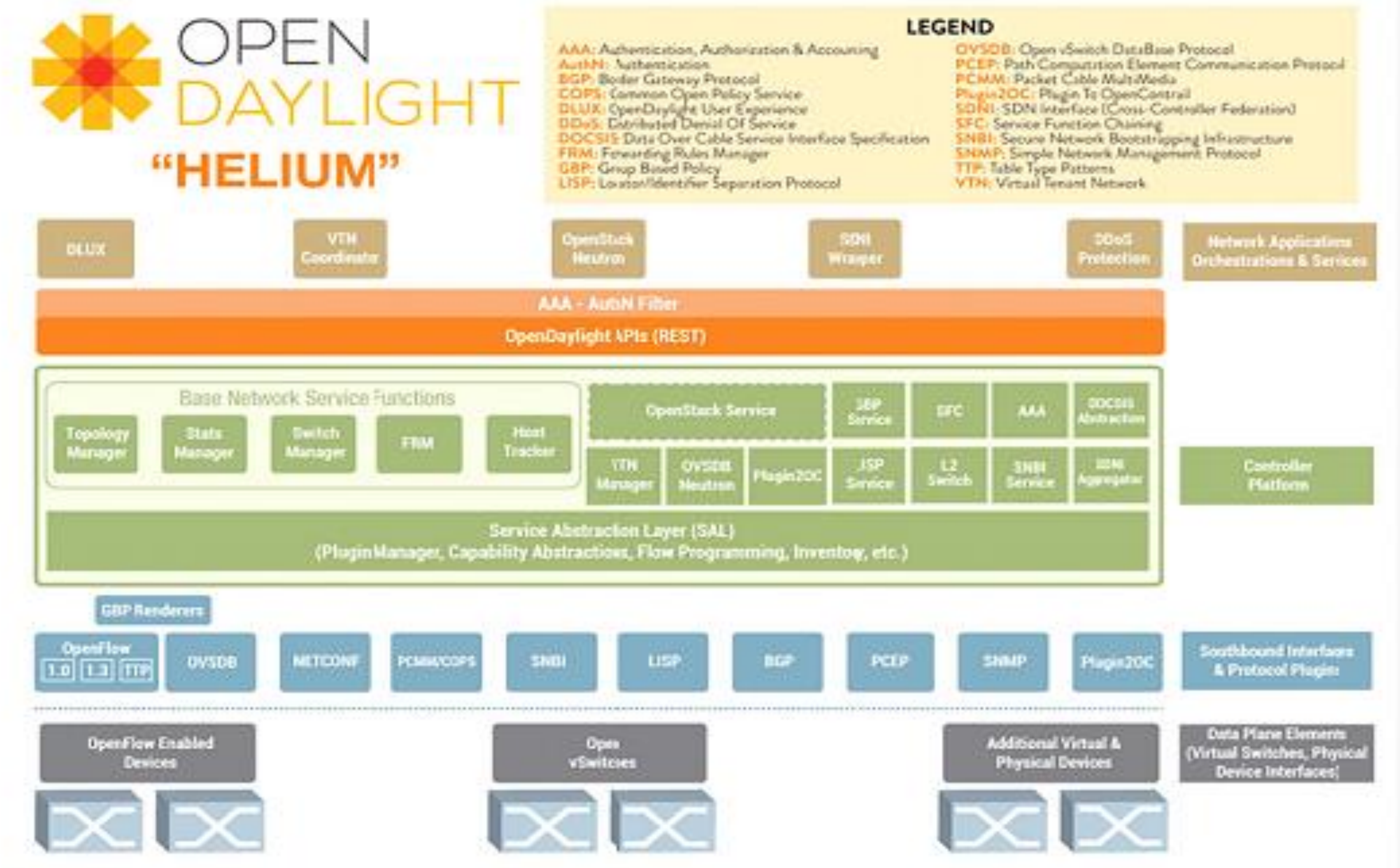
– FlowVisor: a Java OpenFlow controller that behaves like a transparent proxy between an OpenFlow switch and multiple OpenFlow controllers;

– POX: a Python-oriented OpenFlow controller with a high-level SDN interface;

– Floodlight: a Java OpenFlow controller;

– OpenDaylight: OpenDaylight is an open-source project for a modulable platform which contains a controller at its center. The OpenDaylight controller is used as a primary controller by numerous manufacturers, even when those manufacturers have other proprietary solutions that they could use.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

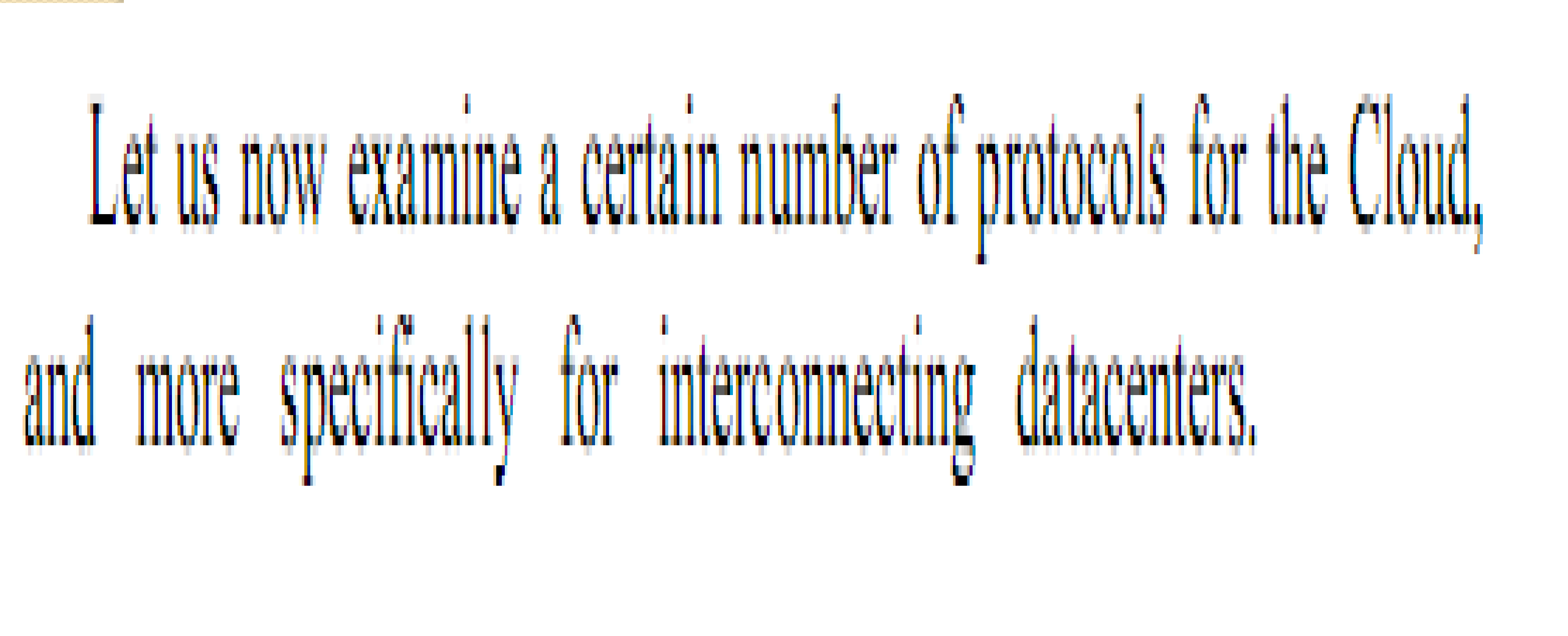


LEGEND

AAA: Authentication, Authorization & Accounting
AuthN: Authentication
BGP: Border Gateway Protocol
COPS: Common Open Policy Service
DLUX: OpenDaylight User Experience
DDoS: Distributed Denial Of Service
DOCSIS: Data Over Cable Service Interface Specification
FRM: Forwarding Rules Manager
GBP: Group Based Policy
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase Protocol
PCEP: Path Computation Element Communication Protocol
PCMM: Packet Cable MultiMedia
Plugin2OC: Plugin To OpenContrail
SDNI: SDN Interface (Cross-Controller Federation)
SFC: Service Function Chaining
SNBI: Secure Network Bootstrapping Infrastructure
SNMP: Simple Network Management Protocol
TTP: Table Type Patterns
VTN: Virtual Tenant Network

OpenDaylight controller

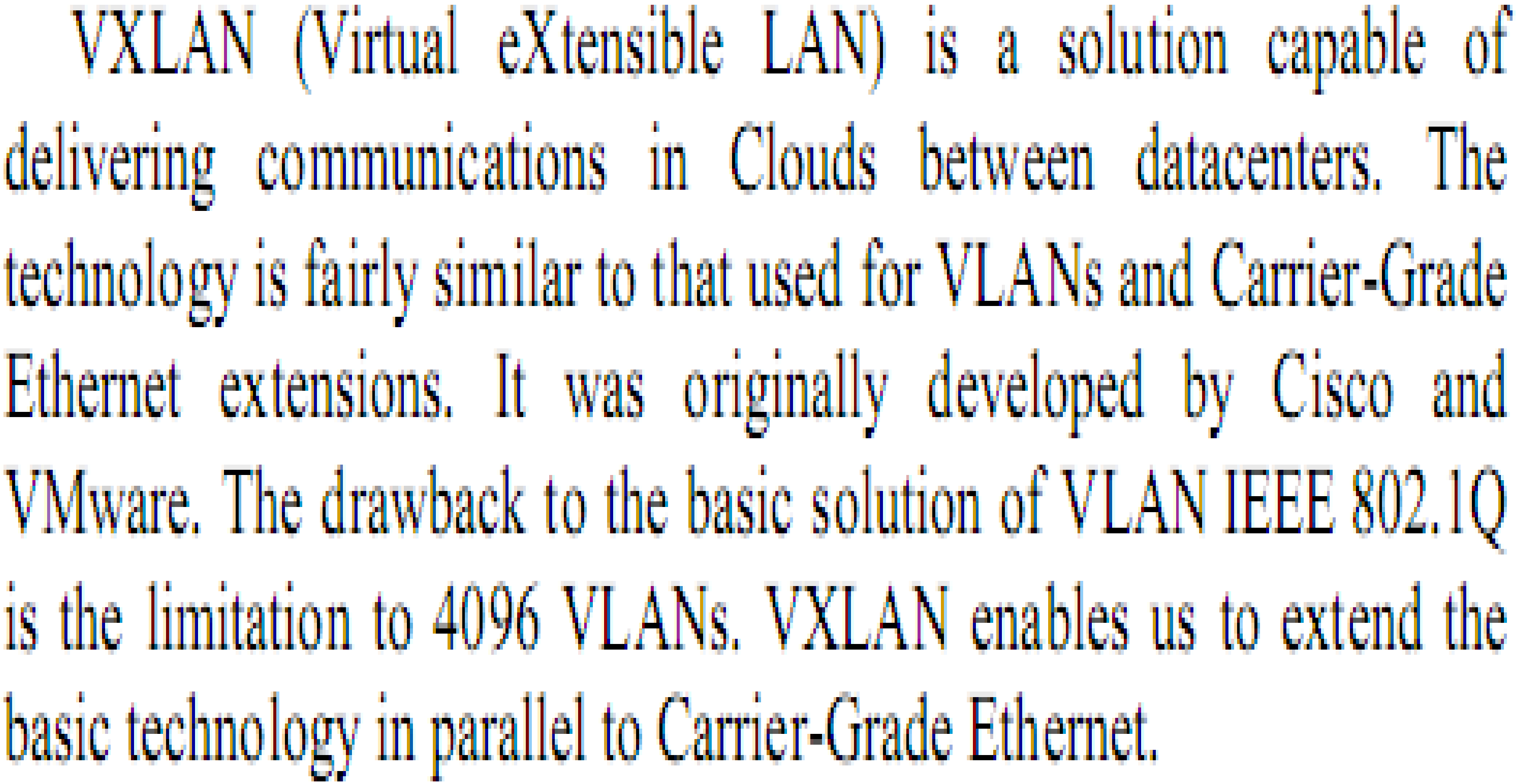
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Let us now examine a certain number of protocols for the Cloud, and more specifically for interconnecting datacenters.

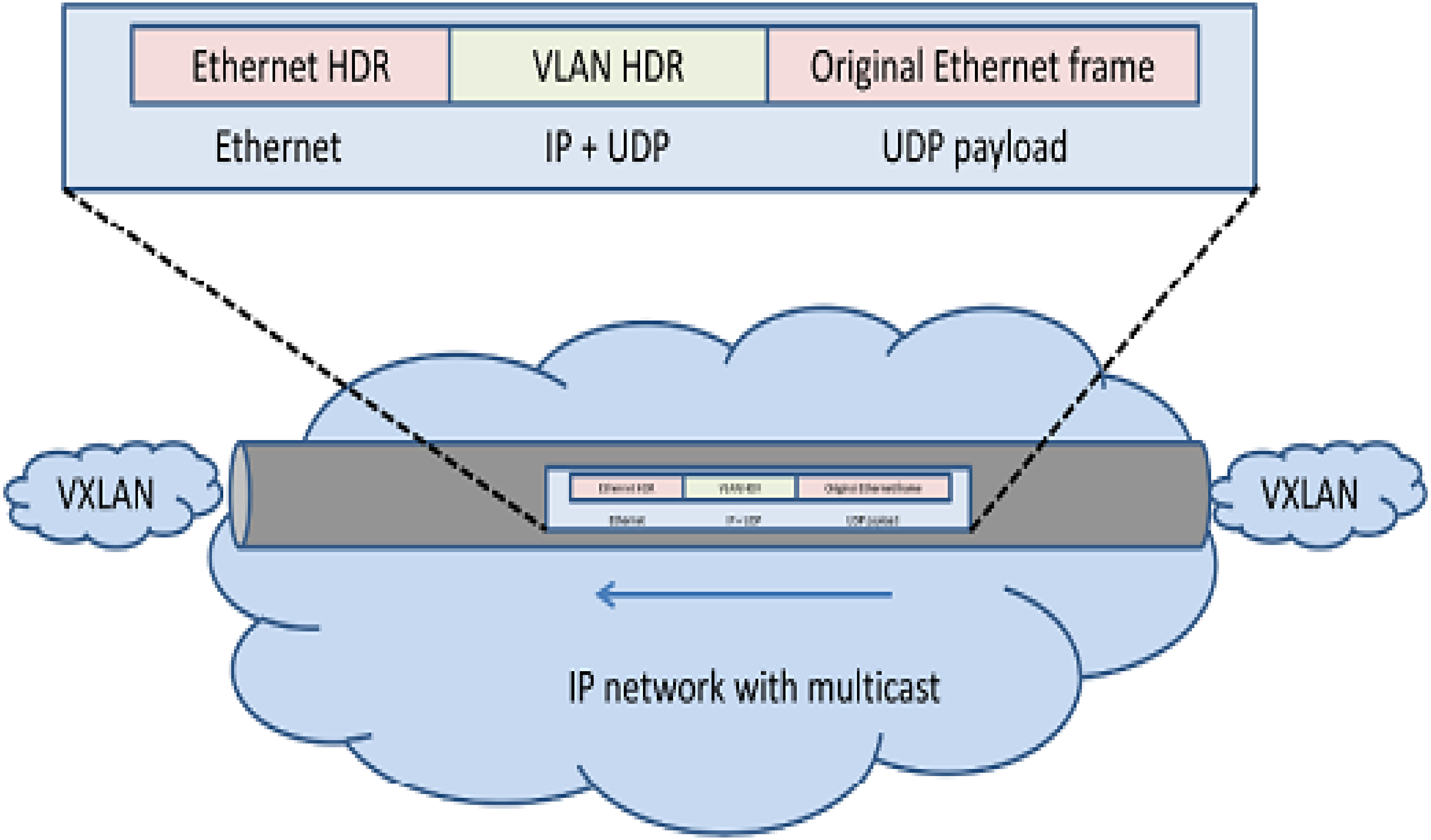
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

VXLAN



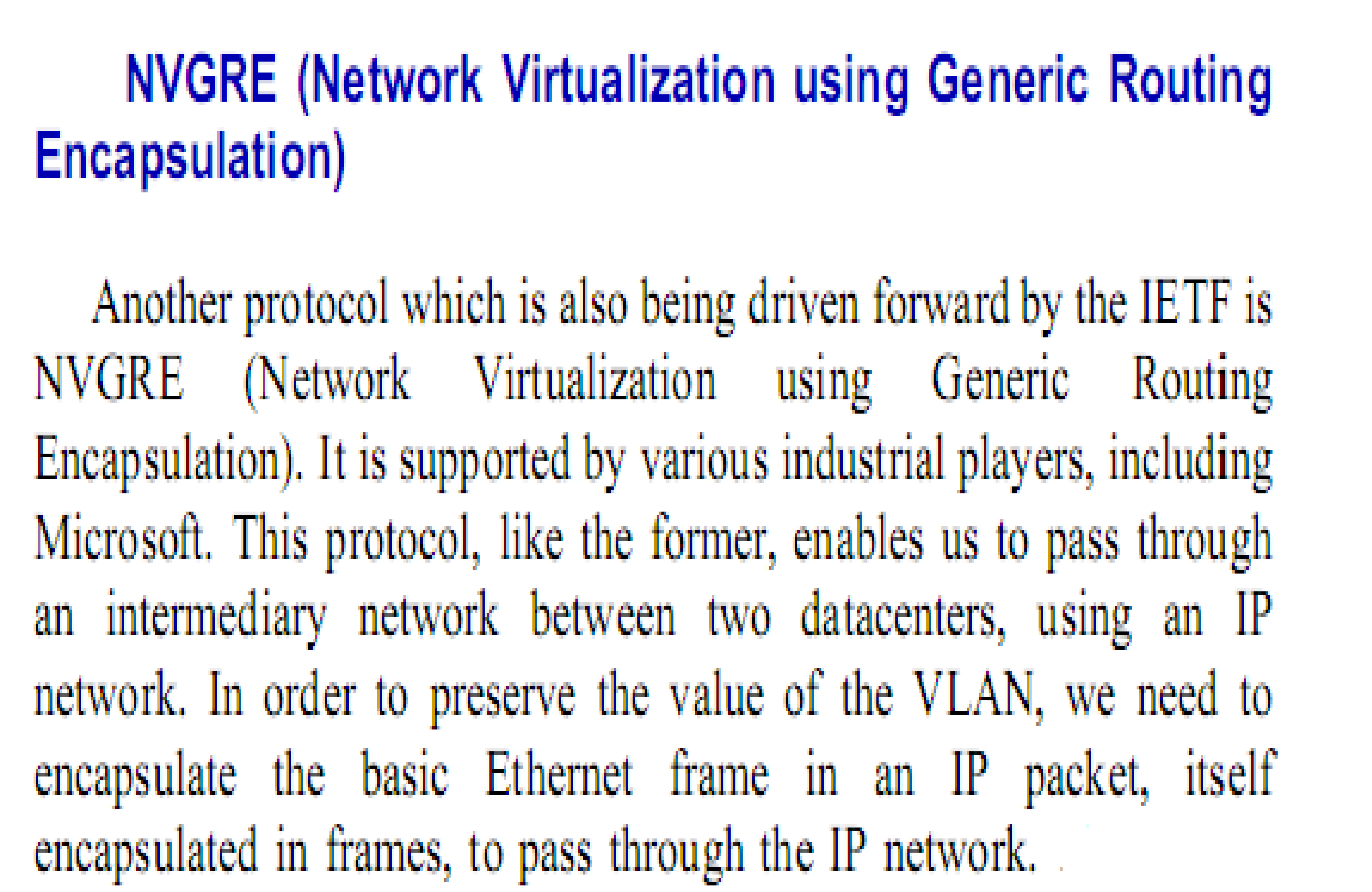
VXLAN (Virtual eXtensible LAN) is a solution capable of delivering communications in Clouds between datacenters. The technology is fairly similar to that used for VLANs and Carrier-Grade Ethernet extensions. It was originally developed by Cisco and VMware. The drawback to the basic solution of VLAN IEEE 802.1Q is the limitation to 4096 VLANs. VXLAN enables us to extend the basic technology in parallel to Carrier-Grade Ethernet.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



VxLAN protocol

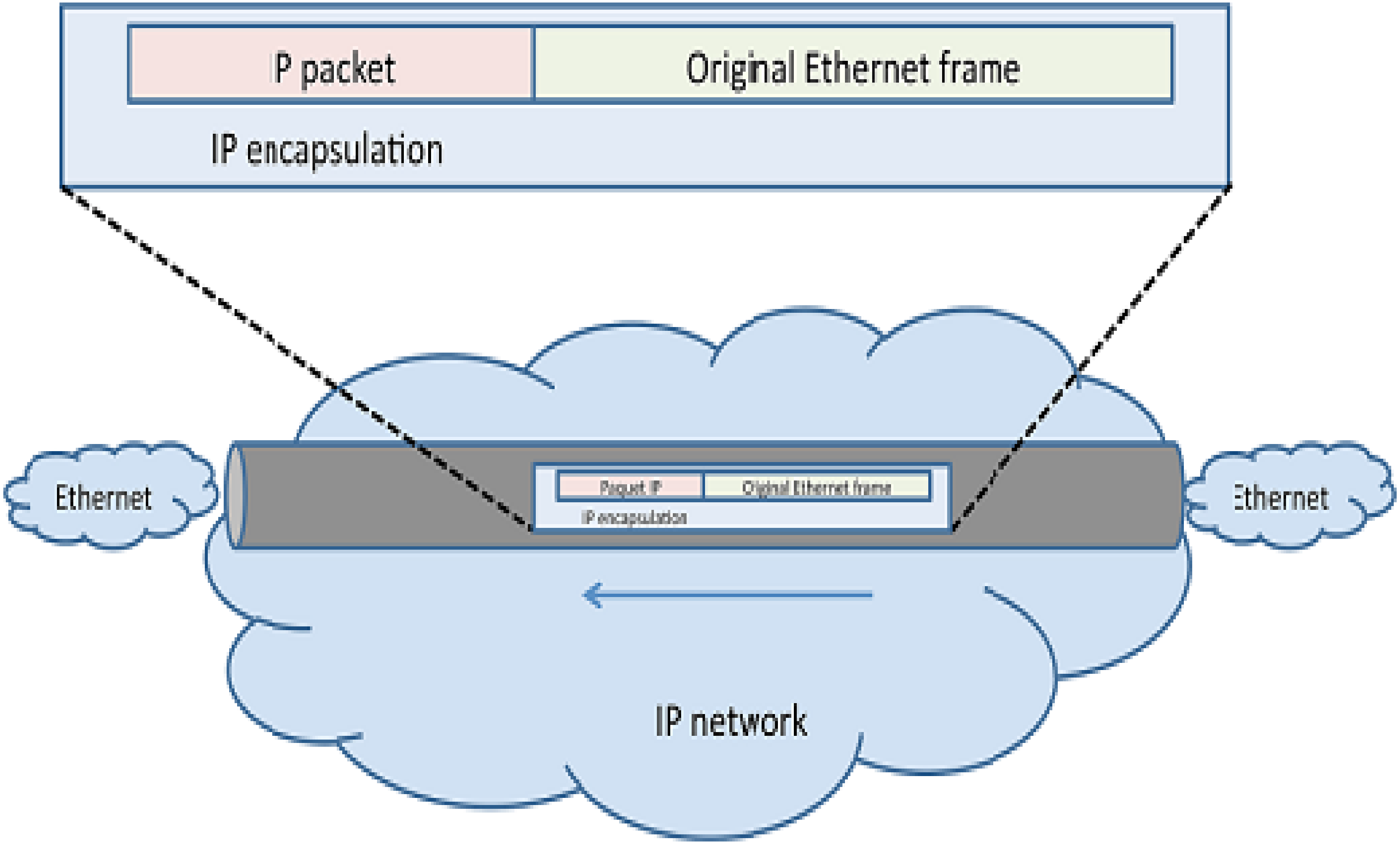
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



NVGRE (Network Virtualization using Generic Routing Encapsulation)

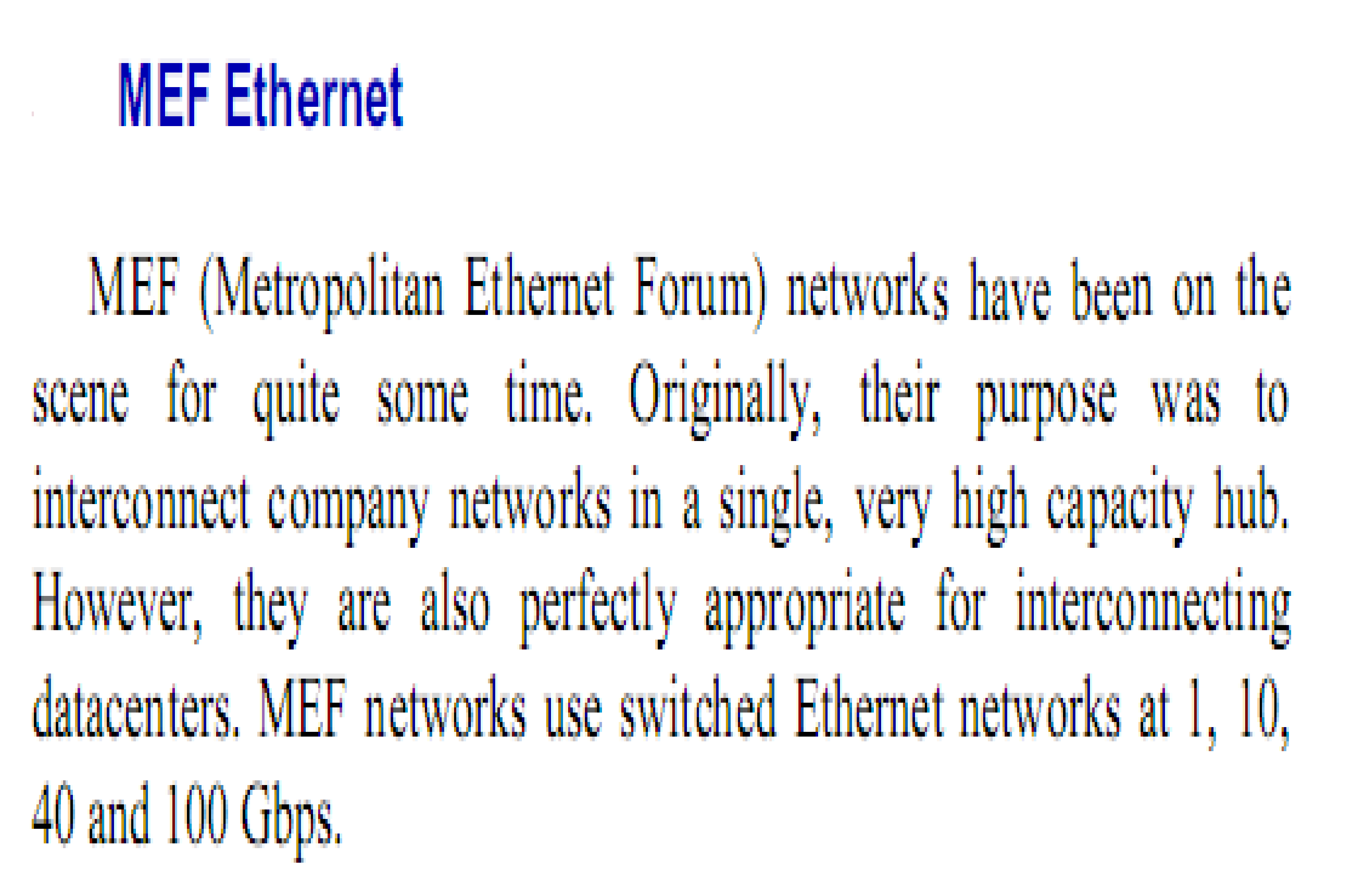
Another protocol which is also being driven forward by the IETF is NVGRE (Network Virtualization using Generic Routing Encapsulation). It is supported by various industrial players, including Microsoft. This protocol, like the former, enables us to pass through an intermediary network between two datacenters, using an IP network. In order to preserve the value of the VLAN, we need to encapsulate the basic Ethernet frame in an IP packet, itself encapsulated in frames, to pass through the IP network.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



NVGRE protocol

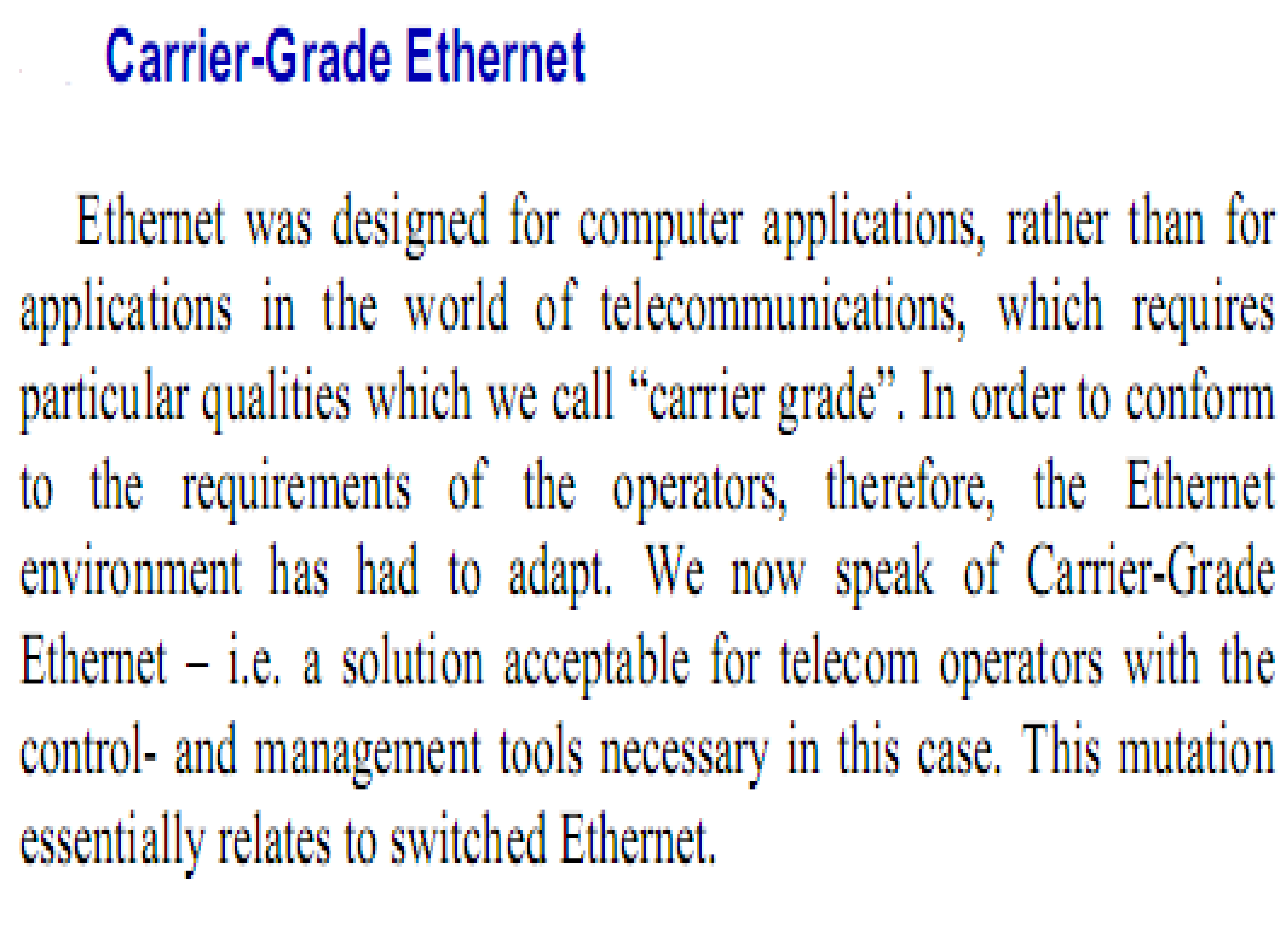
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



MEF Ethernet

MEF (Metropolitan Ethernet Forum) networks have been on the scene for quite some time. Originally, their purpose was to interconnect company networks in a single, very high capacity hub. However, they are also perfectly appropriate for interconnecting datacenters. MEF networks use switched Ethernet networks at 1, 10, 40 and 100 Gbps.

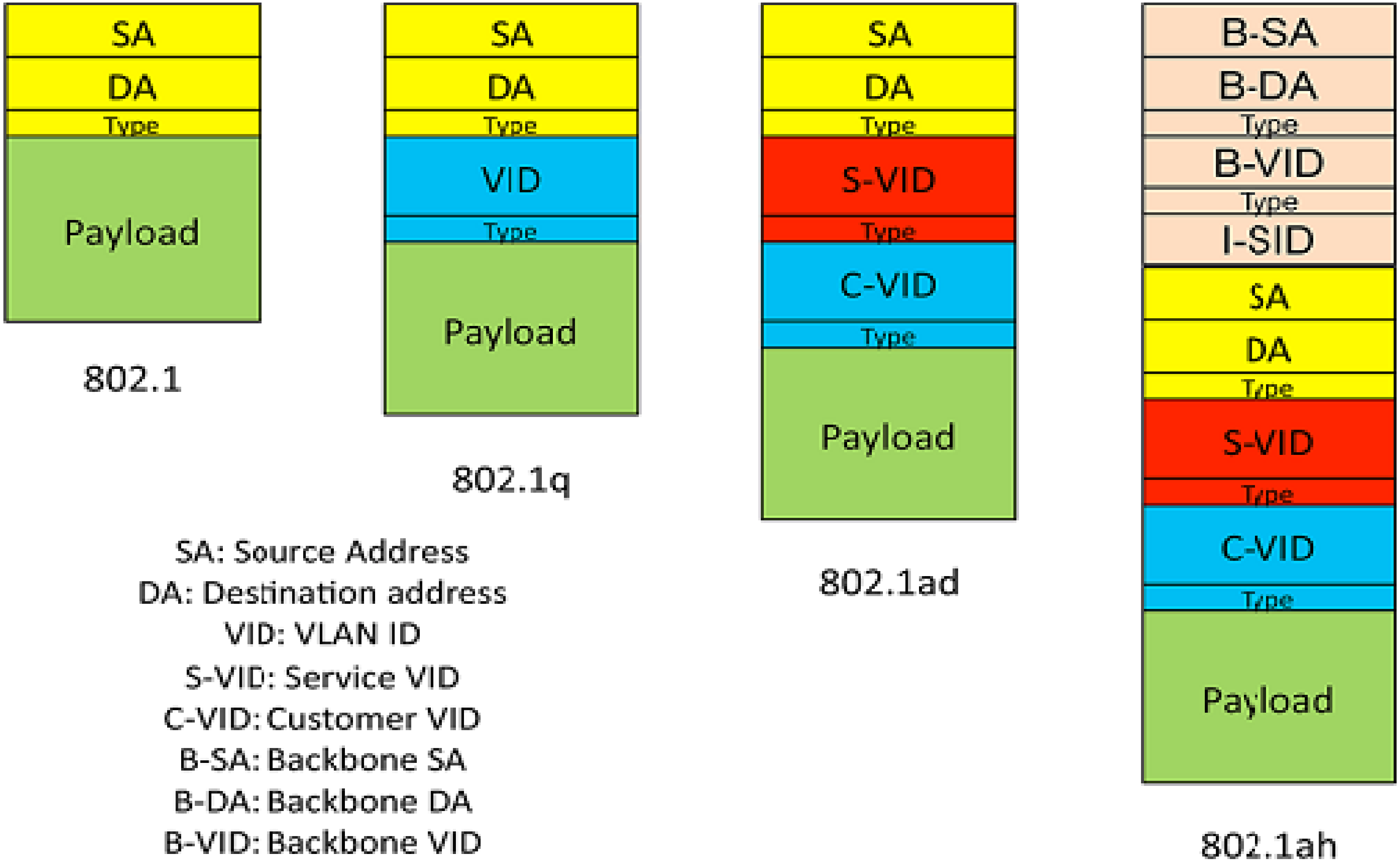
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Carrier-Grade Ethernet

Ethernet was designed for computer applications, rather than for applications in the world of telecommunications, which requires particular qualities which we call “carrier grade”. In order to conform to the requirements of the operators, therefore, the Ethernet environment has had to adapt. We now speak of Carrier-Grade Ethernet – i.e. a solution acceptable for telecom operators with the control- and management tools necessary in this case. This mutation essentially relates to switched Ethernet.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

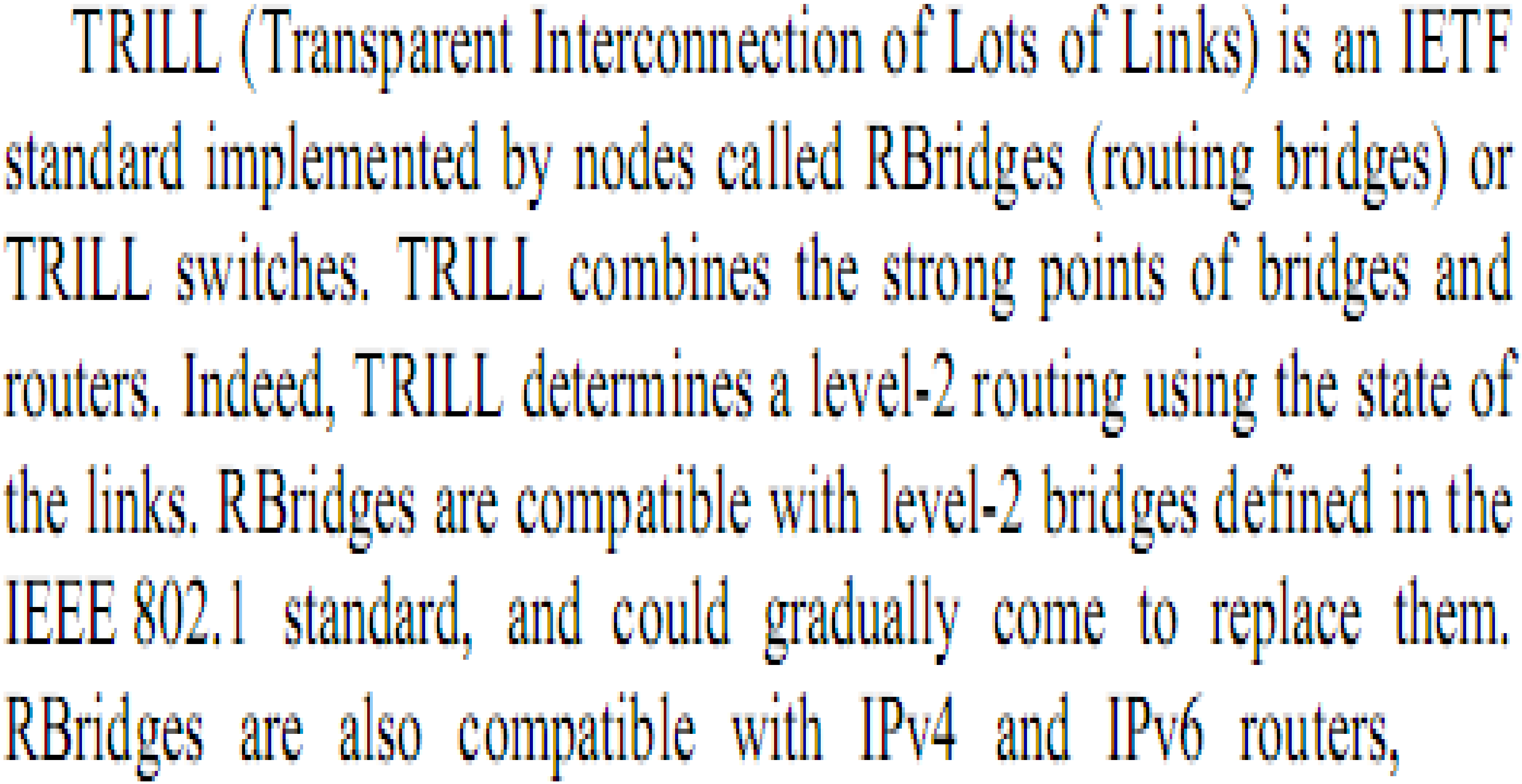


SA: Source Address
DA: Destination address
VID: VLAN ID
S-VID: Service VID
C-VID: Customer VID
B-SA: Backbone SA
B-DA: Backbone DA
B-VID: Backbone VID
I-SID: Service

The different versions of Carrier-Grade Ethernet. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip

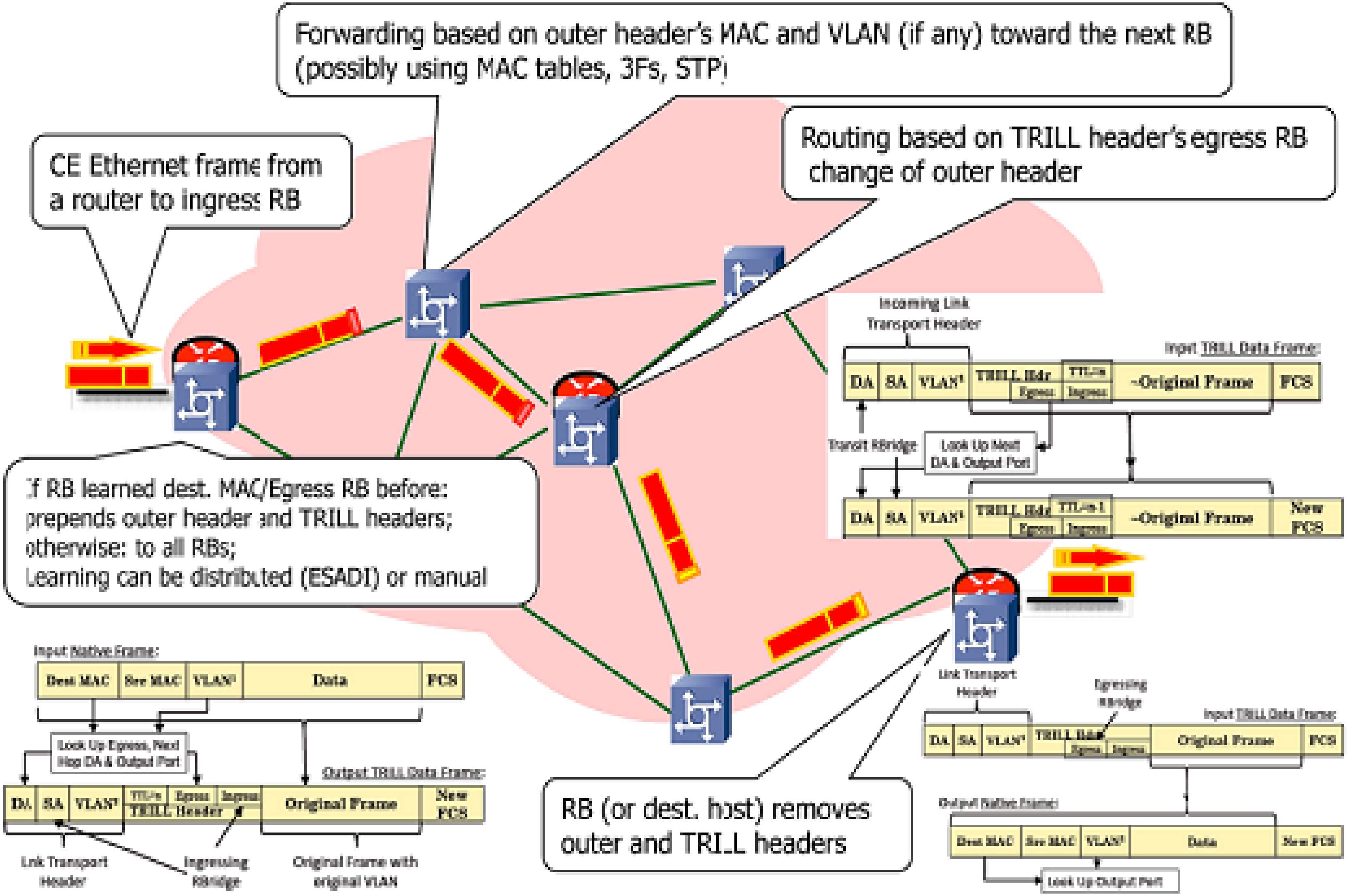
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.

TRILL (Transparent Interconnection of a Lot of Links)



TRILL (Transparent Interconnection of Lots of Links) is an IETF standard implemented by nodes called RBridges (routing bridges) or TRILL switches. TRILL combines the strong points of bridges and routers. Indeed, TRILL determines a level-2 routing using the state of the links. RBridges are compatible with level-2 bridges defined in the IEEE 802.1 standard, and could gradually come to replace them. RBridges are also compatible with IPv4 and IPv6 routers,

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



Forwarding based on outer header's MAC and VLAN (if any) toward the next RB (possibly using MAC tables, 3Fs, STP)

CE Ethernet frame from a router to ingress RB

Routing based on TRILL header's egress RB change of outer header

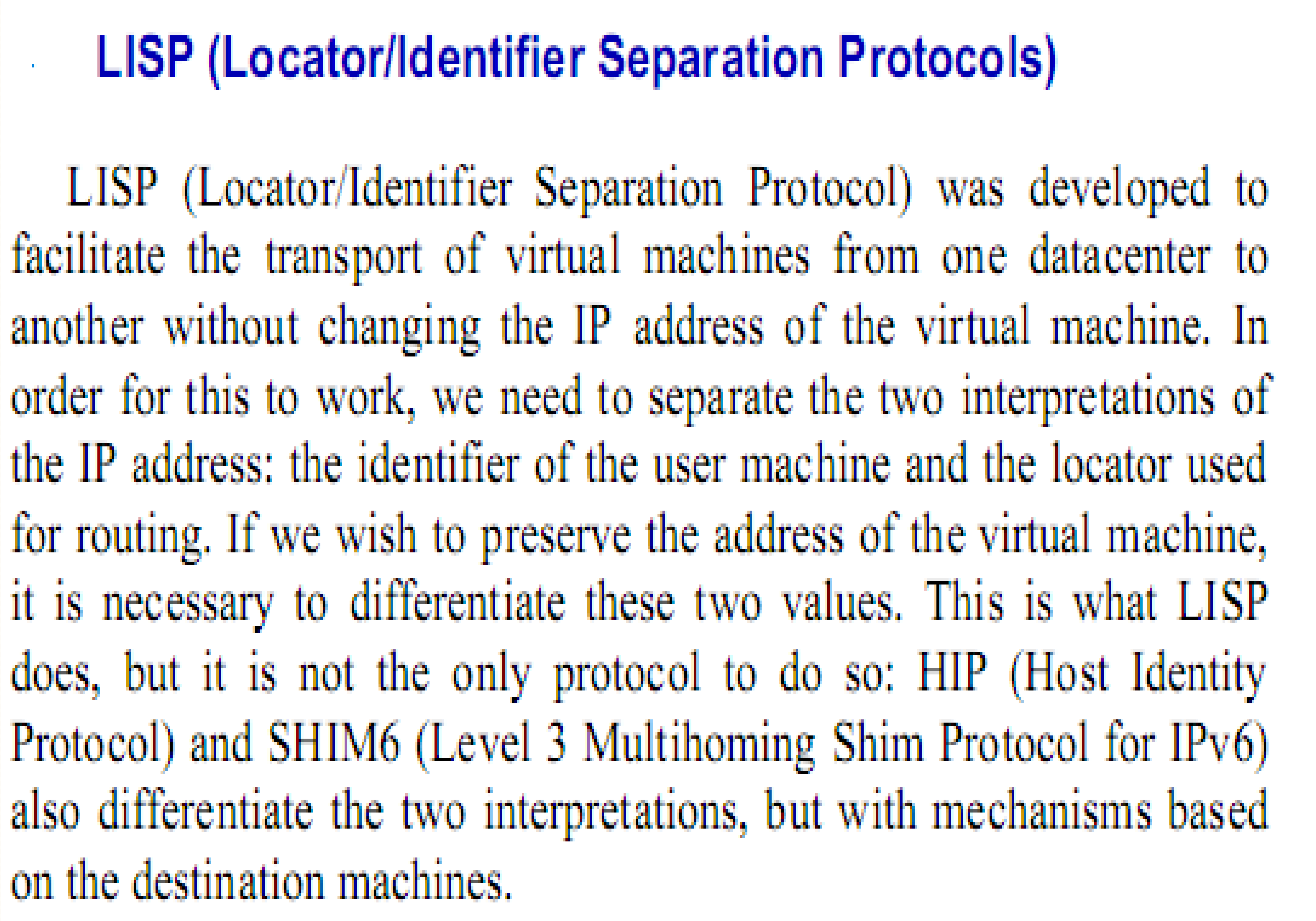
If RB learned dest. MAC/Egress RB before: prepends outer header and TRILL headers; otherwise: to all RBs; Learning can be distributed (ESADI) or manual

RB (or dest. host) removes outer and TRILL headers

Incoming Link Transport Header
Input TRILL Data Frame:
DA | SA | VLAN¹ | TRILL Hdr | TTL=n | Egress | Ingress | Original Frame | FCS
Transit RBridge
Look Up Next DA & Output Port
DA | SA | VLAN¹ | TRILL Hdr | TTL=n-1 | Egress | Ingress | Original Frame | New FCS
Input Native Frame:
Dest MAC | Src MAC | VLAN¹ | Data | FCS
Look Up Egress, Next Hop DA & Output Port
Output TRILL Data Frame:
DA | SA | VLAN¹ | TTL=n | Egress | Ingress | TRILL Header | Original Frame | New FCS
Link Transport Header
Ingressing RBridge
Original Frame with original VLAN
Link Transport Header
Egressing RBridge
Input TRILL Data Frame:
DA | SA | VLAN¹ | TRILL Hdr | Egress | Ingress | Original Frame | FCS
Output Native Frame:
Dest MAC | Src MAC | VLAN¹ | Data | New FCS
Look Up Output Port

TRILL protocol

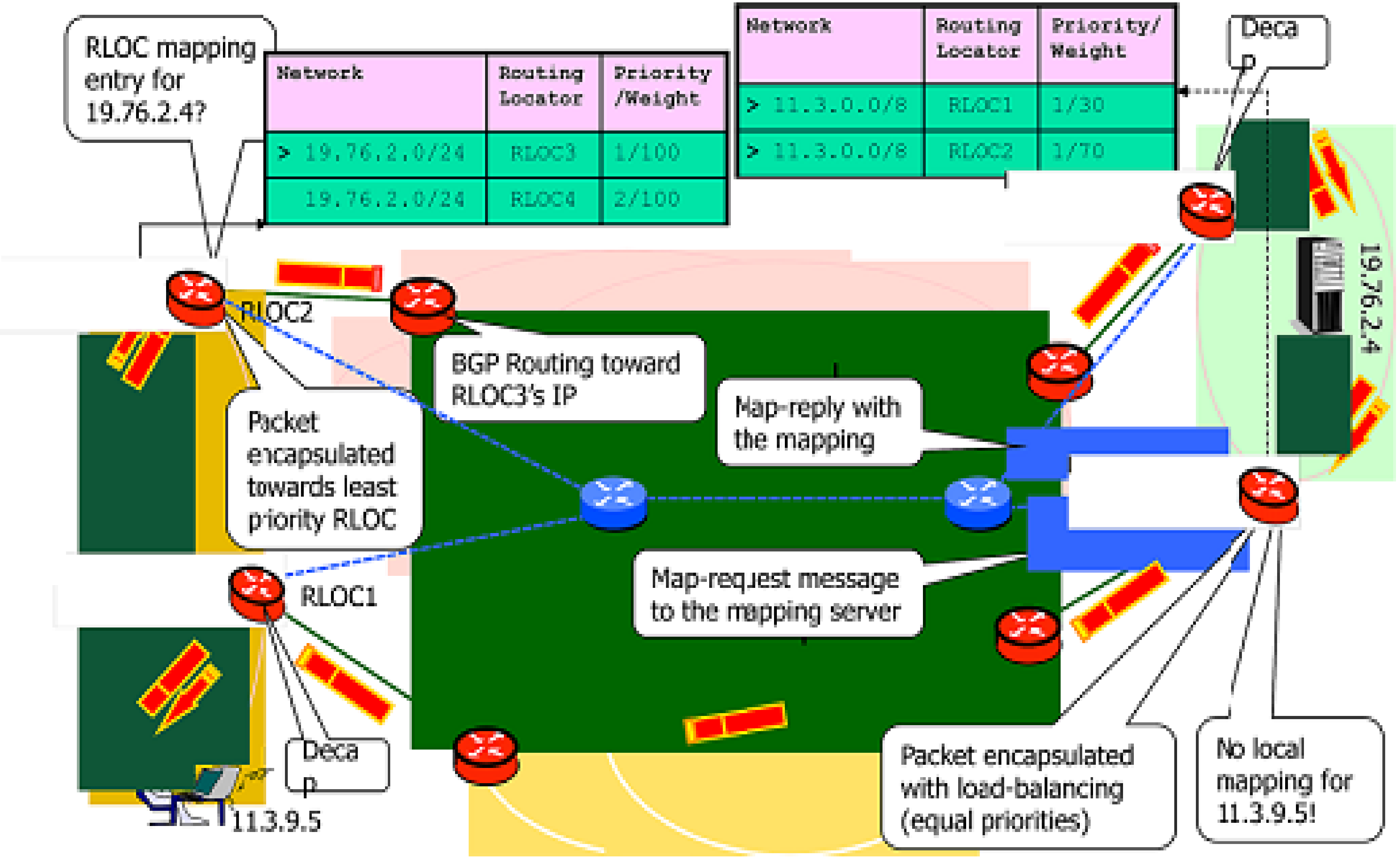
Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



LISP (Locator/Identifier Separation Protocols)

LISP (Locator/Identifier Separation Protocol) was developed to facilitate the transport of virtual machines from one datacenter to another without changing the IP address of the virtual machine. In order for this to work, we need to separate the two interpretations of the IP address: the identifier of the user machine and the locator used for routing. If we wish to preserve the address of the virtual machine, it is necessary to differentiate these two values. This is what LISP does, but it is not the only protocol to do so: HIP (Host Identity Protocol) and SHIM6 (Level 3 Multihoming Shim Protocol for IPv6) also differentiate the two interpretations, but with mechanisms based on the destination machines.

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.



RLOC mapping entry for 19.76.2.4?

Network	Routing Locator	Priority /Weight
> 19.76.2.0/24	RLOC3	1/100
19.76.2.0/24	RLOC4	2/100

Network	Routing Locator	Priority/ Weight
> 11.3.0.0/8	RLOC1	1/30
> 11.3.0.0/8	RLOC2	1/70

BGP Routing toward RLOC3's IP

Map-reply with the mapping

Packet encapsulated towards least priority RLOC

Map-request message to the mapping server

Packet encapsulated with load-balancing (equal priorities)

No local mapping for 11.3.9.5!

LISP protocol

Source : **Guy Pujolle,** Software Networks: Virtualization, SDN, 5G and Security, First Edition, 2015.