A Cloud Optical Access Network for Virtualized GNSS Receivers

ION GNSS+ 2017. Portland, OR, Sept. 27th.

Session F2: GNSS Receiver Processing and Navigation Algorithms 2

C. Fernández-Prades, C. Pomar, J. Arribas,
J.M. Fabrega, J. Vilà-Valls, M. Svaluto Moreolo,
R. Casellas, R. Martínez, M. Navarro, F.J. Vílchez,
R. Muñoz, R. Vilalta, L. Nadal, A. Mayoral

Fundació Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)

Overview

1. Context.
2. Proposed system architecture.
3. Proof of concept.
4. Conclusions.

Context

Motivation

- In recent years, the Information Technologies (IT) industry has witnessed an explosion of successful new business models, services and applications based on the combination of three key concepts: the cloud, Software Defined Networking and Network Function Virtualization.
- The application of such concepts and associated technologies allows companies to increase the efficiency and flexibility of their IT resources by managing them as logical entities instead of as physical, hardwired units dedicated to a given application or service.
- How can those concepts be applied in the framework of GNSS receivers?

Three key concepts in the IT industry

- **The Cloud**
  - A term referring to accessing compute, storage and software applications through a network connection,
  - often by accessing data centers using wide area networking (WAN) or Internet connectivity.
- **Software Defined Networking (SDN)**
  - Centralized Network Control model allowing network administrators to programmatically initialize, control, change, and manage network behavior dynamically via open interfaces.
- **Network Function Virtualization (NFV)**
  - Architecture and interfaces for logically dividing IT resources to implement virtual network node functions (i.e., functional entities that act as physical devices) which can then be managed as building blocks that may be connected, or chained together, to create communication services.

Examples of Virtualization: Network appliances

FROM:

- Fragmented, non-commodity dedicated hardware.
- Physical installation per appliance per site.

Message Router

Session Border Controller

Firewall

WAN acceleration

Content Delivery Network

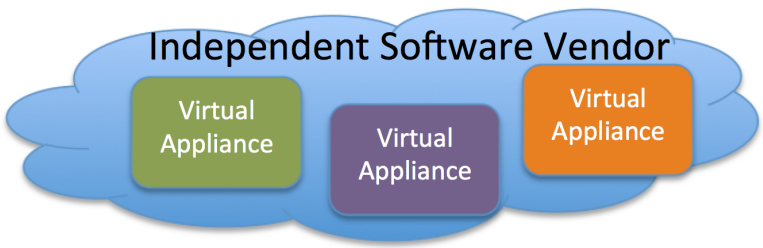
Tester

Radio Network Controller

Carrier-grade NAT

TO:

- Orchestrated, automatic, remote installation and operation.



Standard high-volume Computers

Standard high-volume Switches

Standard high-volume Storage

Proposed concept

To replace physical GNSS receivers by a *Virtual Network Function (VNF)*, which can be automatically managed and deployed in minutes.

- Introduction of the concept of virtualized GNSS receiver.
  - Natural extension of software-defined GNSS receivers.
  - The user just needs the antenna and network connectivity.
  - The GNSS receiver (RF + baseband) is elsewhere.
  - Automated end-to-end service orchestration.
- Based on ultimate trends and tools from the IT industry.
- Enabling new applications and services based on GNSS signals.
- It still poses some technological challenges.

The Good: This is a service enabler

- A network of GNSS reference stations can produce differential data to provide real-time corrections (high accuracy) to third users, or to store digitized signals for future analysis.
- Programmable output rate of GNSS observables, paving the way to new scientific applications (*e.g.*, space weather, geodesy).
- Controlled reception pattern antennas (CRPAs): Robustness to jamming / multipath. Localization of interference sources.
- Rapid deployment of GNSS-related infrastructure.
- Convenient solution for GNSS Commercial Services, GNSS Authentication, and security-related applications (M-code, PRS), since the encryption module remains on the service provider's premises.
- ... and many more!

The Bad: Technological challenges

- **Availability**: targeting 24/7 service. This means:
  - High bandwidth, low latency, and possibly QoS and encryption mechanisms in the transport network.
  - High computational capabilities for the cloud infrastructure: the host device executing the software-defined GNSS receiver must reach real-time.
- **Scalability**:
  - How such a system adapts to more users, more GNSS signals and bands, more external data sources or to more complex signal and data processing algorithms.
- **Reliability**:
  - A trusted receiver for certification / security aspects.
- **Efficiency**:
  - Power consumption trade-off between the user equipment and the cloud infrastructure.
- Others: interoperability, marketability, etc.

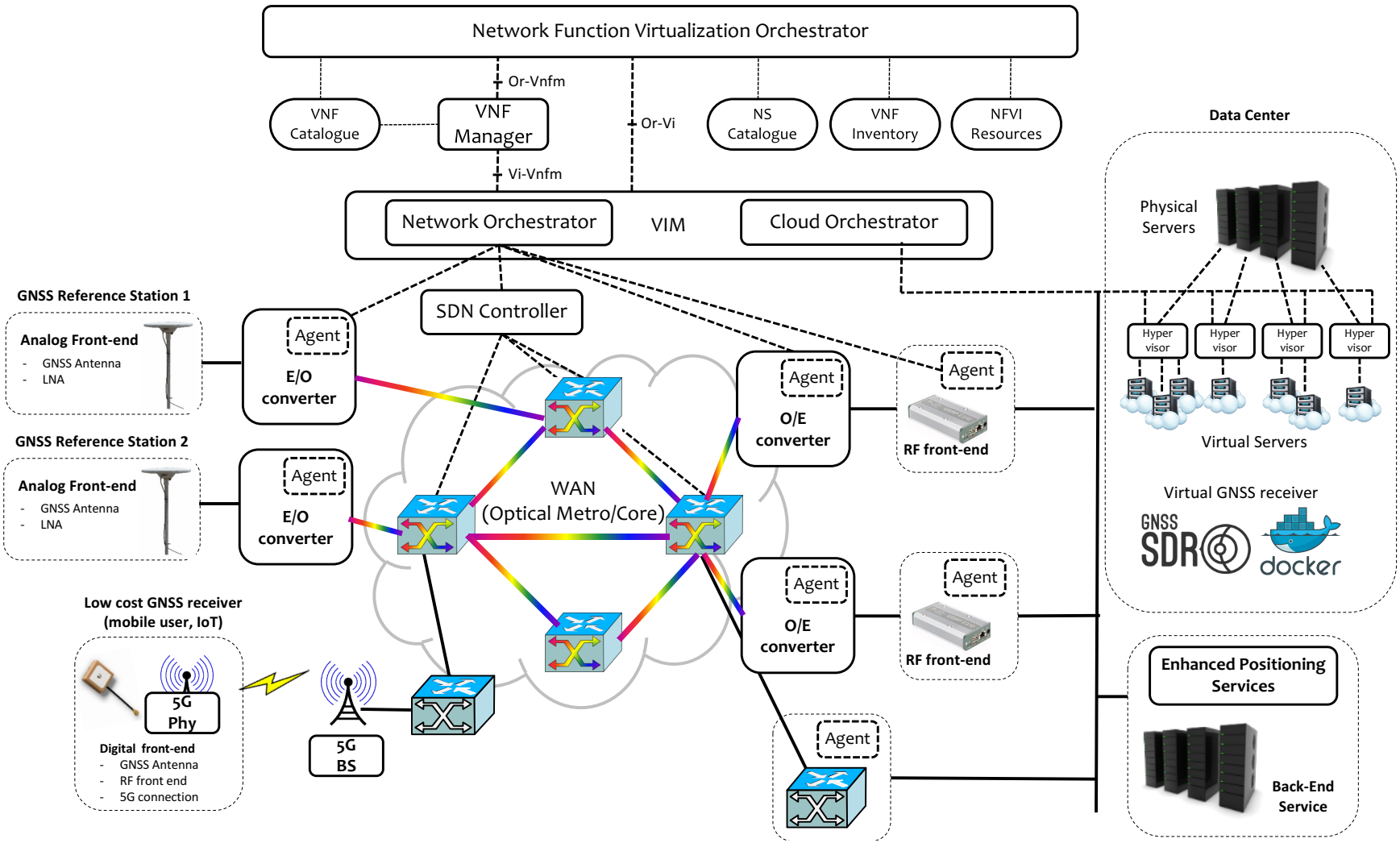
The Ugly: Required throughput

Band	Freq. (MHz)	Service	GNSS Signal	Min. required bandwidth BW_{min}	Min. bit rate with $q = 2$ at BW_{min}	Min. bit rate with $q = 8$ at BW_{min}
E5	1176.45	OS/SoL	GPS $L5^{(*)}$	20.46 MHz	81.84 Mbps	327.36 Mbps
	1176.45	OS	Galileo E5a	20.46 MHz	81.84 Mbps	327.36 Mbps
	1176.45	OS	GLONASS $L5OCM^{(**)}$	N/A; ~20 MHz	~80 Mbps	~320 Mbps
	1191.795	PRS	Galileo E5	N/A; **~50 MHz**	~200 Mbps	**~800 Mbps**
	1202.025	OS	GLONASS $L3OC^{(*)}$	20.46 MHz	81.84 Mbps	327.36 Mbps
	1207.14	OS/SoL	Galileo E5b	20.46 MHz	81.84 Mbps	327.36 Mbps
	1207.14	OS	Beidou B2I	N/A; ~4 MHz	16 Mbps	64 Mbps
	1207.14	M	Beidou B2Q	N/A; ~20 MHz	~80 Mbps	~320 Mbps
	1207.14	OS	GLONASS $L3OCM^{(**)}$	N/A; ~20 MHz	~80 Mbps	~320 Mbps
L2	1227.60	OS	GPS $L2C^{(*)}$	2.046 MHz	8.184 Mbps	32.736 Mbps
	1227.60	M	GPS P	20.46 MHz	81.84 Mbps	327.36 Mbps
	1227.60	M	GPS $M^{(*)}$	N/A; ~24 MHz	~96 Mbps	~384 Mbps
	1246.00	OS	GLONASS L2OF	6.7095 MHz	26.838 Mbps	107.352 Mbps
	1246.00	M	GLONASS L2SF	15.907 MHz	63.63 Mbps	254.52 Mbps
	1248.06	OS	GLONASS $L2OC^{(**)}$	N/A; ~4 MHz	~16 Mbps	~64 Mbps
	1248.06	M	GLONASS $L2SC^{(**)}$	N/A; ~15 MHz	~60 Mbps	~240 Mbps
E6	1268.52	M	Beidou B3	20.46 MHz	81.84 Mbps	327.36 Mbps
	1278.75	CS	Galileo E6b/c	10.23 MHz	40.920 Mbps	163.68 Mbps
	1278.75	PRS	Galileo E6a	N/A; ~30 MHz	~120 Mbps	~480 Mbps
L1	1561.098	OS	Beidou B1	4.092 MHz	16.368 Mbps	65.472 Mbps
	1575.42	OS	GPS C/A	**2.046 MHz**	8.184 Mbps	**32.736 Mbps**
	1575.42	M	GPS P	20.46 MHz	81.84 Mbps	327.36 Mbps
	1575.42	OS	GPS $L1C^{(**)}$	4.092 MHz	16.368 Mbps	65.472 Mbps
	1575.42	M	GPS $M^{(*)}$	N/A; ~24 MHz	~96 Mbps	~384 Mbps
	1575.42	OS	GLONASS $L1OCM^{(**)}$	N/A; ~ 4 MHz	~16 Mbps	~64 Mbps
	1575.42	PRS	Galileo E1a	N/A; ~40 MHz	~160 Mbps	~640 Mbps
	1575.42	OS/SoL	Galileo E1b/c	4.092 MHz	16.368 Mbps	392.832 Mbps
	1600.995	OS	GLONASS $L1OC^{(**)}$	N/A; ~ 4 MHz	~16 Mbps	~64 Mbps
	1600.995	M	GLONASS $L1SC^{(**)}$	N/A; ~15 MHz	~60 Mbps	~240 Mbps
	1602.00	OS	GLONASS L1OF	8.3345 MHz	33.338 Mbps	133.352 Mbps
	1602.00	M	GLONASS L1SF	17.5325 MHz	70.13 Mbps	280.52 Mbps

(*): Modernized signal not broadcast by all satellites. (**): Proposed signal. Count ~20% more due to transport overhead

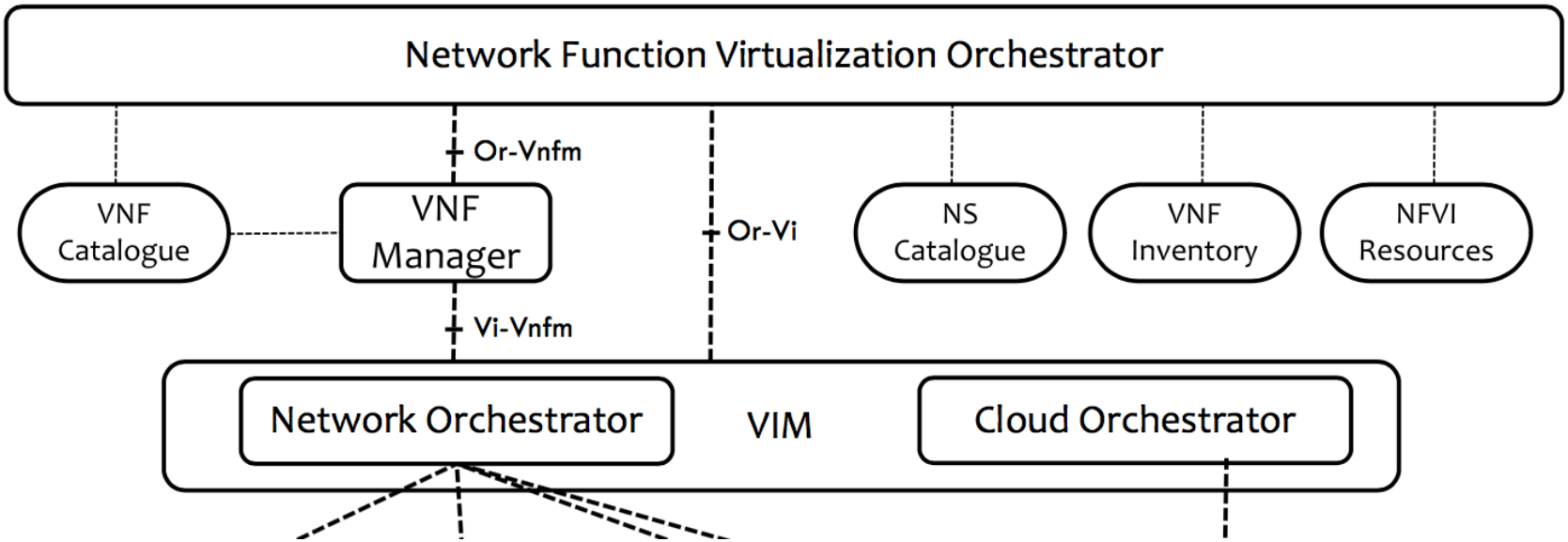
Proposed System Architecture

End-to-end NFV management and orchestration



Based on the architecture proposed by NFV MANO, a working group of the European Telecommunications Standards Institute (ETSI).

Network Function Virtualization Orchestrator

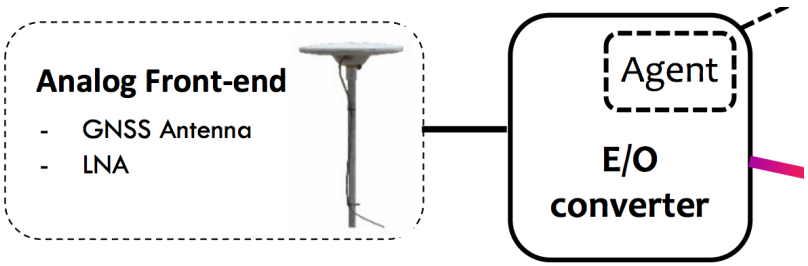


- **NFV Orchestrator:** service on-boarding, service lifecycle management, authorization of resource requests.
- **Virtualized Network Function Manager**: lifecycle of VNF instances (instantiation, scaling in/out, scaling up/down, upgrading, termination).
- **Virtualized Infrastructure Manager (VIM)**
  - Network orchestrator: end-to-end communication.
  - Cloud orchestrator: manages virtual IT resources.

VNF: Virtual Network Function.

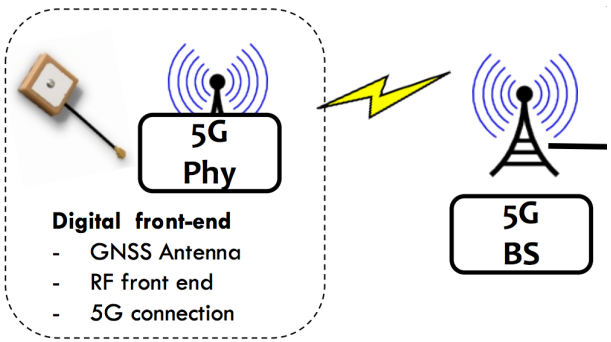
User Equipment

Type A: Static users with connectivity to an optical network.



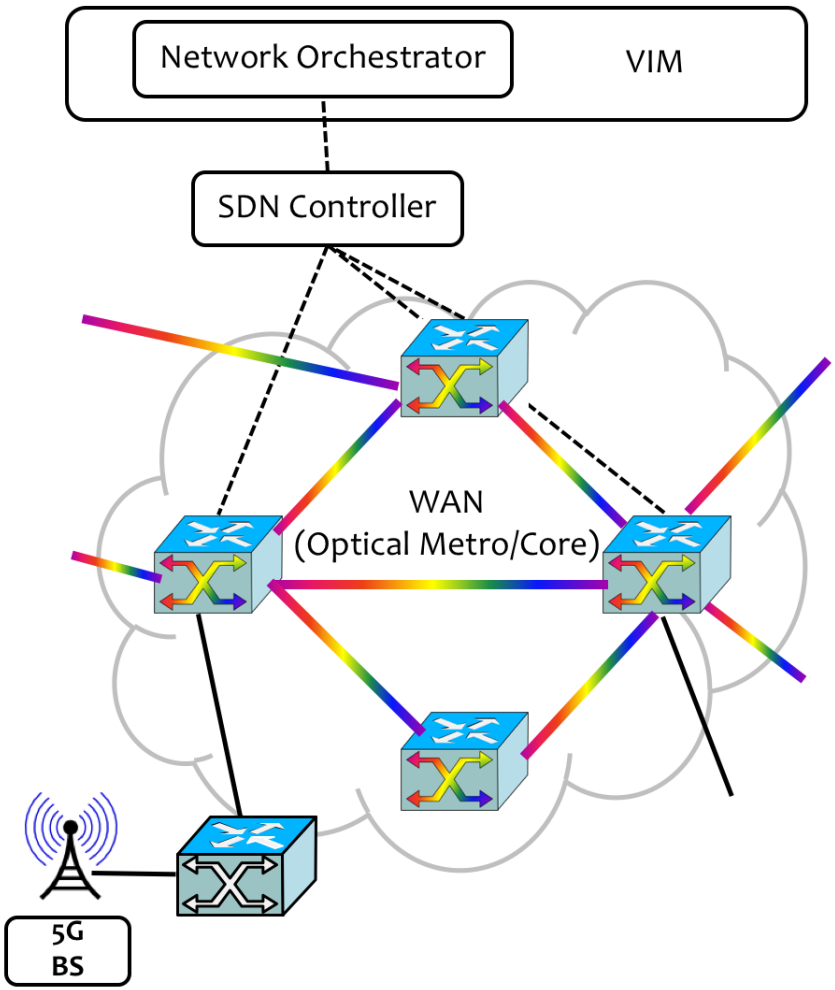
- One or more antennas.
- Electrical-to-Optical (E-O) converter.
- Connection to an optical network (radio-over-fiber).

Type B: Mobile users (or users without connection to an optical network)



- One or more antennas.
- RF front-end(s) (downconversion + ADC).
- **Wireless interface** (maybe in 5G?). Batches of data.

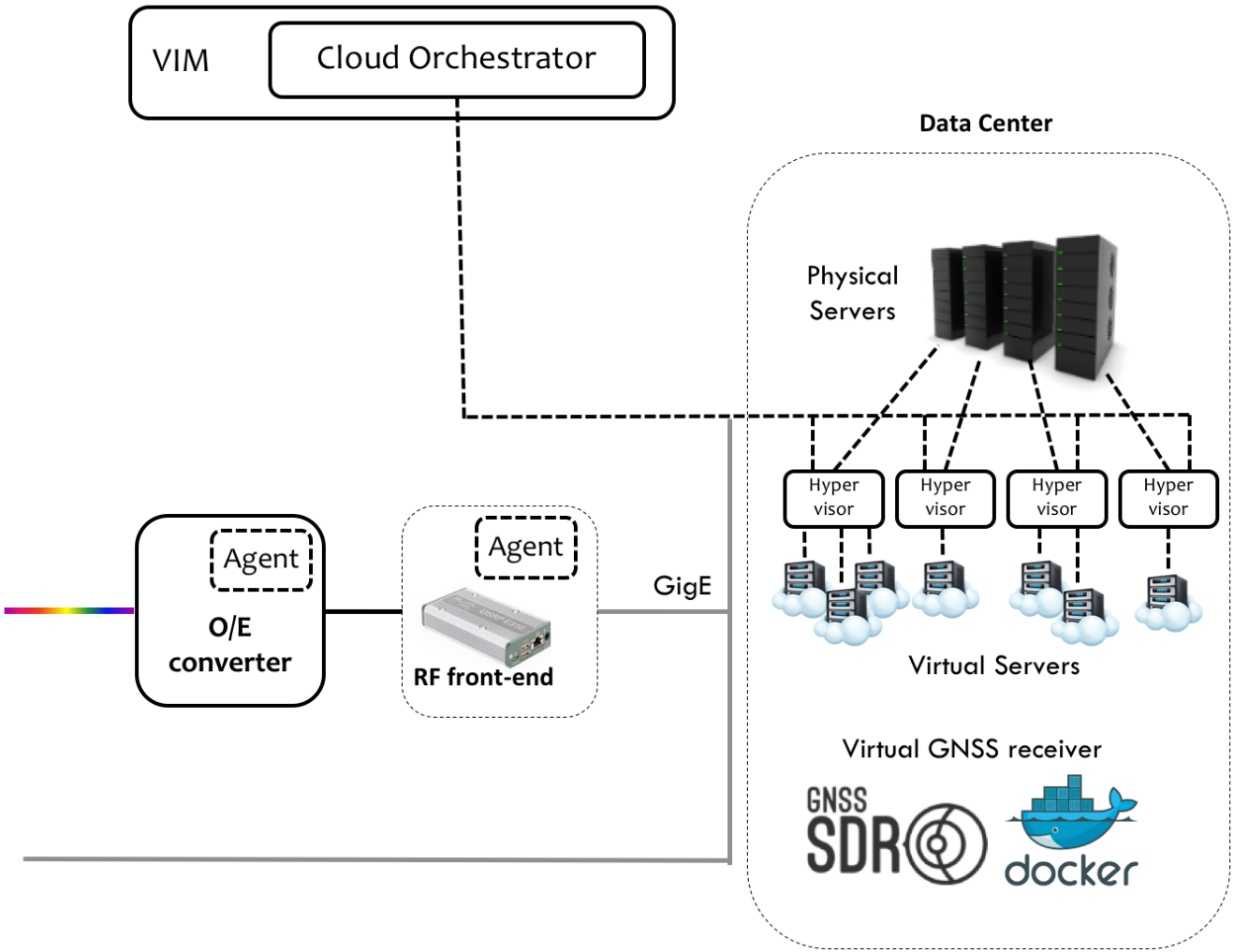
Software Defined Optical Network



- **SDN Controller**:
  - Enables application ecosystem and business logic "above".
  - Provides end-to-end connectivity, configuring flow control of Network Elements "below".
- **Network elements**:
  - Packet (e.g. L2) switches
  - Optical switches such as Reconfigurable optical add/drop multiplexers (ROADM) avoid O-E-O conversion.

Back-end

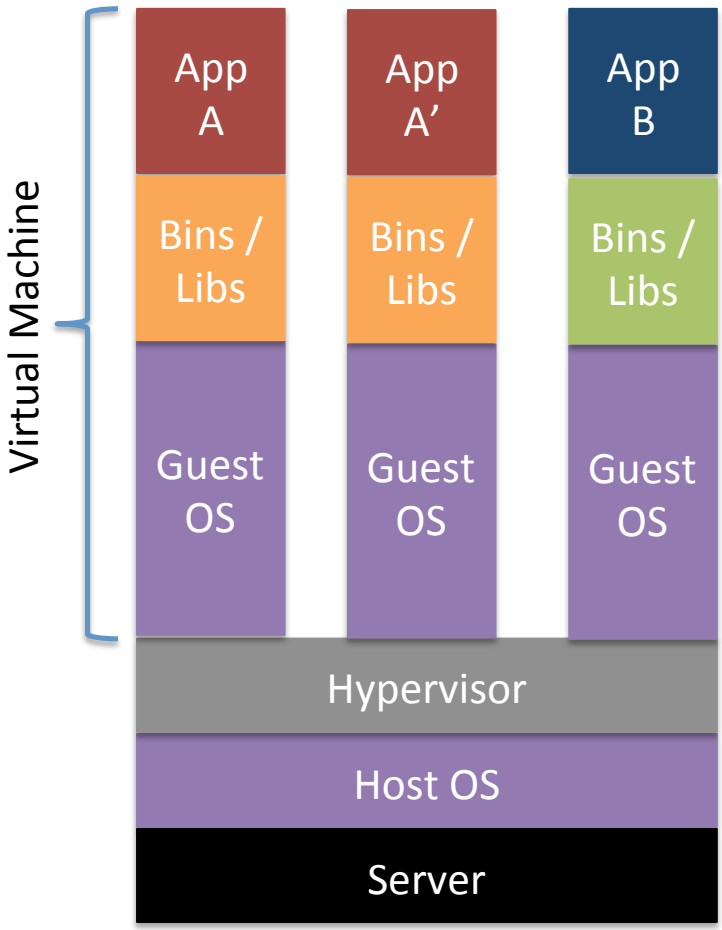
A data center with compute, storage and network resources.

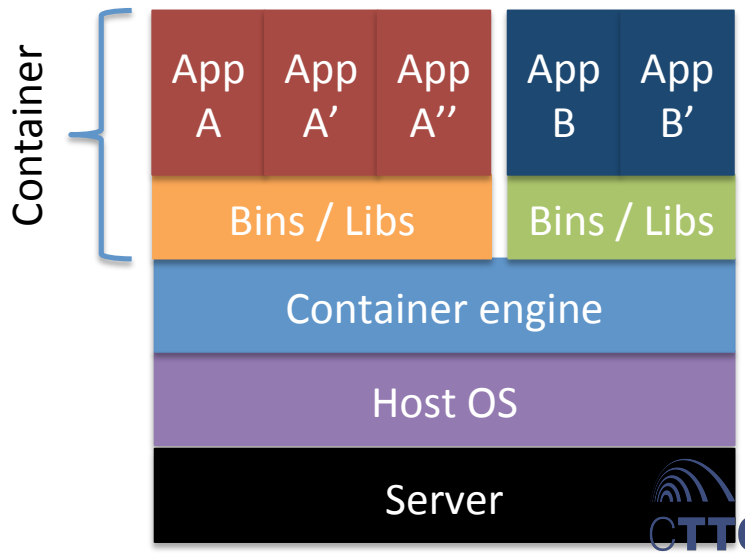


The virtualized GNSS receiver

A software-defined GNSS receiver can be virtualized:

- As a **Virtual Machine**: full software stack.
- As a **container**: lighter and more efficient.

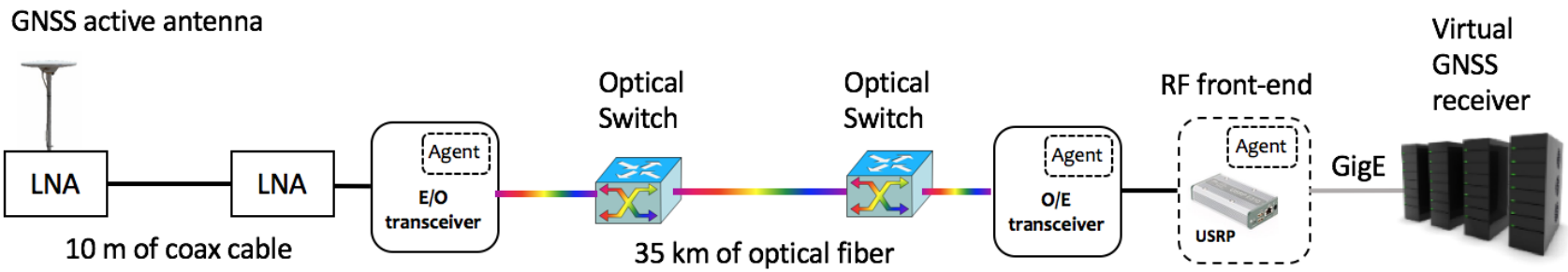




Proof Of Concept

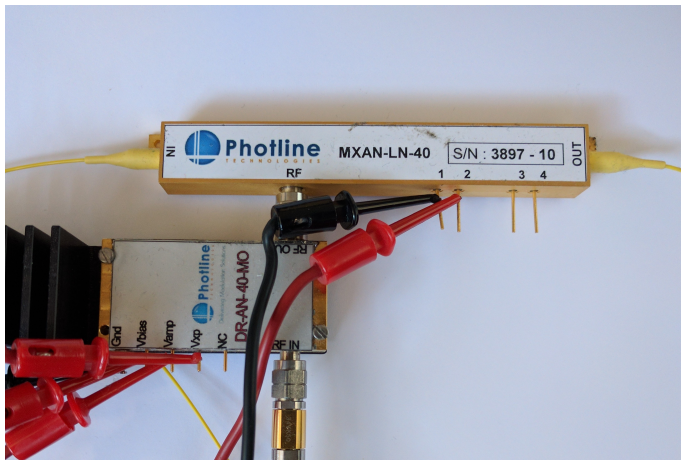
Experiment design

- KISS approach.
- Static user, one antenna, with optical fiber connectivity.
- Simple optical network configuration.
- All implemented with free and open source software and commercial-off-the-shelf hardware components.
- End-to-end demonstration (from the antenna to position in real-time and with live GNSS signals, in a realistic environment).



Experimental setup (I/II)

Antenna platform

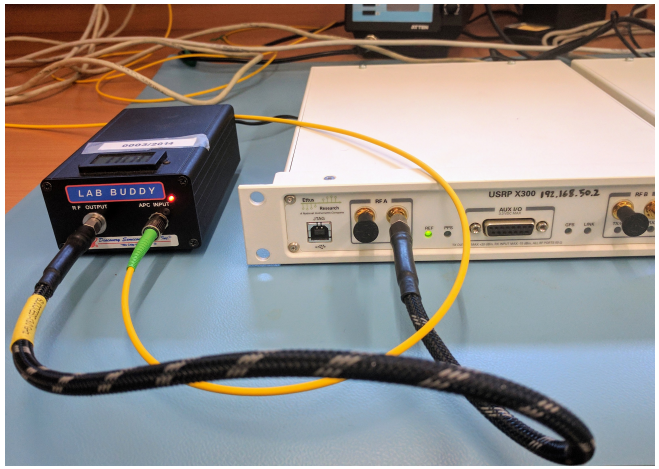


E-O converter

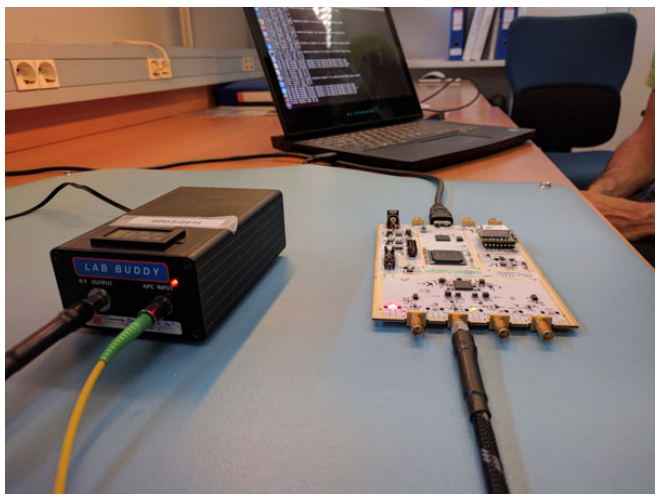
ADRENALINE Testbed®

35 km of optical fiber

Experimental setup (II/II)



O-E converter (left) and RF front-end (right).



More COTS options.

Open source everywhere

Managed by **OpenStack** with:

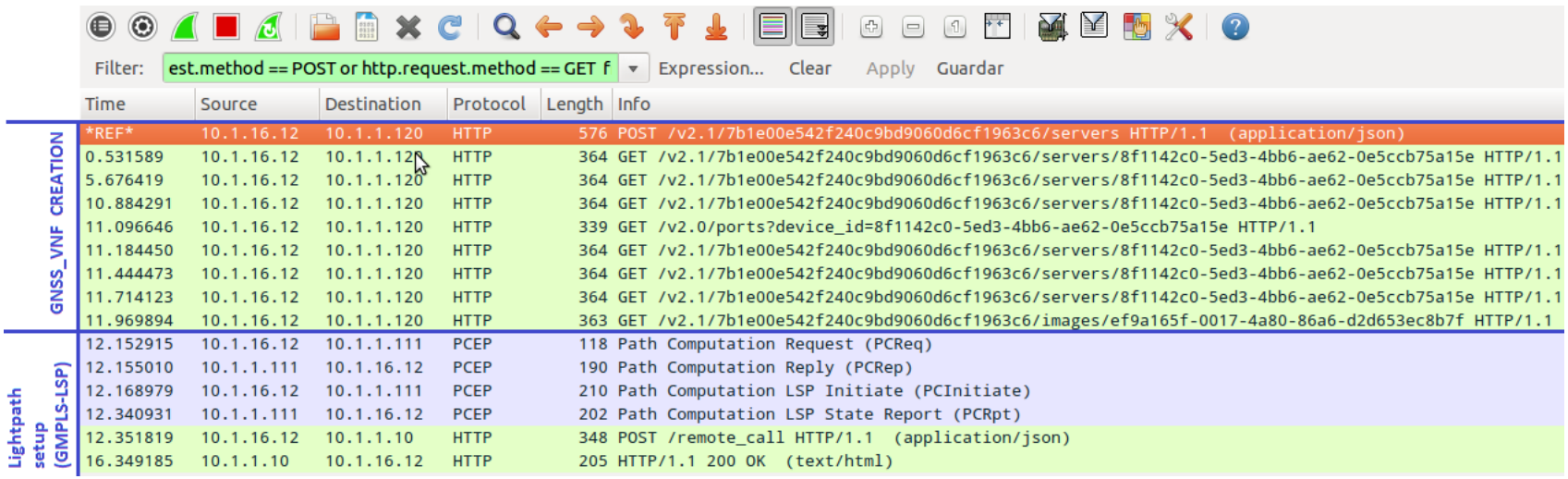
- **Keystone** (authentication)
- **Nova** (compute resources)
- **Neutron** (network connectivity within the DC)

3 virtualization technologies:

1. **GNSS-SDR** running in a virtual machine prepared with **VirtualBox**;
2. **GNSS-SDR** in a **Docker container**;
3. **GNSS-SDR** in an Ubuntu's **Snap package**.

Experimental results (I/III)

System deployment in less than 20 s.



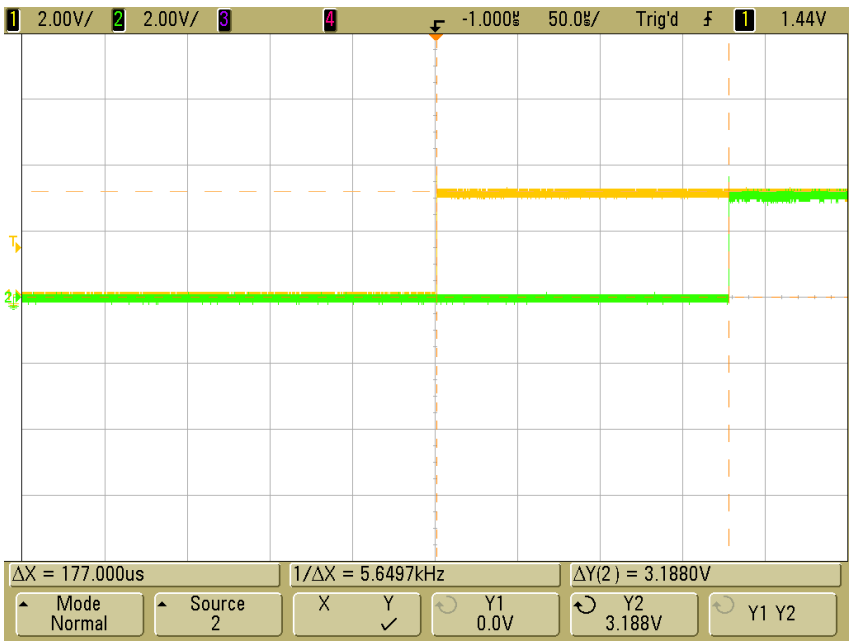
Network packet traces as captured by Wireshark, a network protocol analyzer.

- The creation of the GNSS virtual network function took **11.9 s**.
- The lightpath setup in the optical network took **4.2 s**.

Experimental results (II/III)

Low and stable latency

End-to-end latency measured with two commercial GNSS receivers.



The **latency** measured from the antenna to the output of the O-E converter at the back-end was **177 μs**, from which **116 μs** correspond to the light propagation time along 35 km.

Experimental results (III/III)

All three virtualization mechanisms attained real-time processing and got PVT fixes using live GPS L1 C/A and Galileo E1 Open Service signals.

- ✓ GNSS-SDR in a Virtual Machine. We used Virtual Box, and Ubuntu 16.04 as Host and Guest OS.
- ✓ GNSS-SDR on a Docker container.
  `https://github.com/carlesfernandez/docker-gnsssdr`
- ✓ GNSS-SDR on a Snap Package
  `https://uappexplorer.com/snap/ubuntu/gnss-sdr-next`

Conclusions

Conclusions (I/II)

- This work introduced the concept of **virtualized GNSS receivers**, in the context of network function virtualization for optical access networks.
- We identified some **technological bottlenecks**.
- We presented a **new system architecture** based on latest industry standards for end-to-end management and orchestration of virtual network functions.
- We reported the results of a **proof-of-concept** that demonstrates the feasibility of the physical separation between the GNSS antenna and the virtual GNSS receiver.

Conclusions (II/II)

- Future work:
  - Scalability: more signals, more bandwidth, more users.
  - In deep analysis of VM vs Containers.
  - Latency monitoring.
- An open source virtualized GNSS receiver packed as a Docker container is freely available at `https://github.com/carlesfernandez/docker-gnsssdr`
- Snap package freely available from the Ubuntu Store: `https://uappexplorer.com/snap/ubuntu/gnss-sdr-next`

Thank you for your attention!