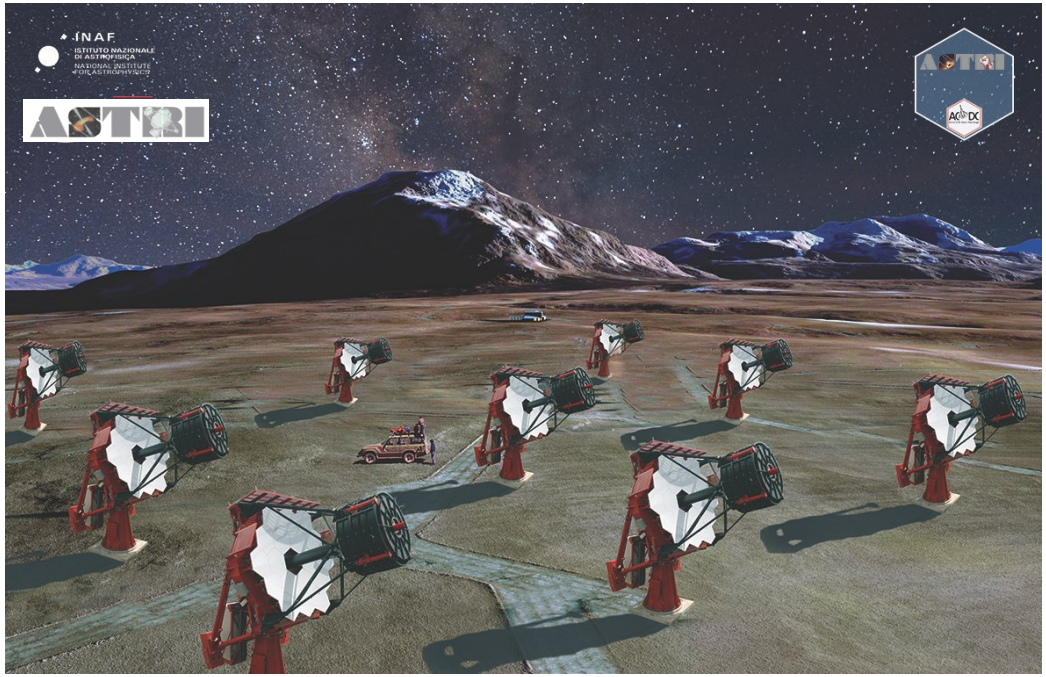
ASTRI Astrofisica con Specchi a Tecnologia Replicante Italiana



ASTRI Virtual Test Bed: from Prototype to Mini-Array

Fulvio Gianotti

for the ASTRI Project

INAF/OAS-Bologna, Italy

This work was conducted in the context of the ASTRI Project

OUTLINE

- ASTRI Test Bed (TB) for SW Development
- Existing Test Bed Based on Oracle VM
  - ASTRI Test Bed Hardware
  - ASTRI Test Bed Architecture
  - TB support Operating System Upgrade
- HW and SW limitations of the current TB
- The choice of the ProxMox VE virtualization system
- ProxMox VE: Learning the system
- HW architecture of a ProxMox VE System for the ASTRI Mini-Array
- Conclusion and Developments

ASTRI SST-2M telescope prototype

ASTRI SST-2M, designed to comply with all CTA-SST requirements, is an 'end-to-end' telescope prototype: it comprises all of the work that should be done to achieve the final scientific products.

ASTRI Prototype is installed in the in SLN INAF-OACT Observatory on Etna Mountain

- Telescope
  - Structure
  - Mirrors geometry and coating
- Camera
  - Photosensors and electronics
  - Thermal system
  - Ancillary devices
  - Camera Control
  - Data acquisition
- Calibration
  - Camera calibration
  - External equipment for pointing and calibration
- Control system
  - Tracking and pointing
  - Monitoring and alarm
- Data reduction and analysis
  - Pipelines
  - Data archiving
- ICT Infrastructure
  - Complete and stand alone Computing Center

Is Impossible to integrate and test the SW on the Prototype ICT

ASTRI Test Bed reproduces the SLN Prototype environment

- **Network Services**: Firewall, NAT, VPN, DNS, LDAP, Frontier Server (**astrisln**), ISCSI/NFS Storage=> 2 Server
- **Telescope Control and Monitor Servers**: slntmcdb, slnics, slntcs, slnomc, slnaux, => 5 Server
- **Data Acquisition, Archiving and Analysis Servers:** slndaq, slncluster1, slnstorage.=> 3 Server

Why ASTRI Test bed

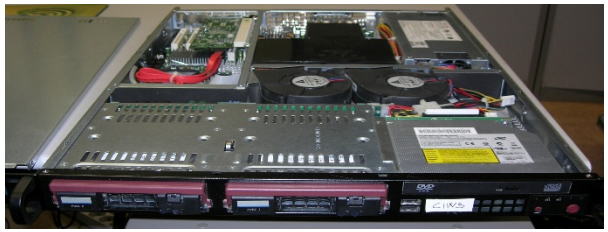
- software integration and test
- Continuous integration with Jenkins
- Distributed ACS system configuration test

Test Bed as virtual environment

- The Virtualization System is less expensive than "real" system in term of Cost, Time and Power
- It can run more than 20 VMs in 3 physical servers
- The virtual Machines are easily replicable
- A copy of VM has been distributed to the developers
- Test Bed is based on Oracle VM
- Oracle VM system is fast and reliable and free for small installations
- The OVM system can ensure high availability
- Oracle VM allows to have a single control console to easily manage multiple servers OVMS and dozens of Virtual Machines.

2013-2014 Implementation

OVM MANAGER



Internet/
OAS LAN

Management LAN 1Gbit

OVMS1

OVMS3

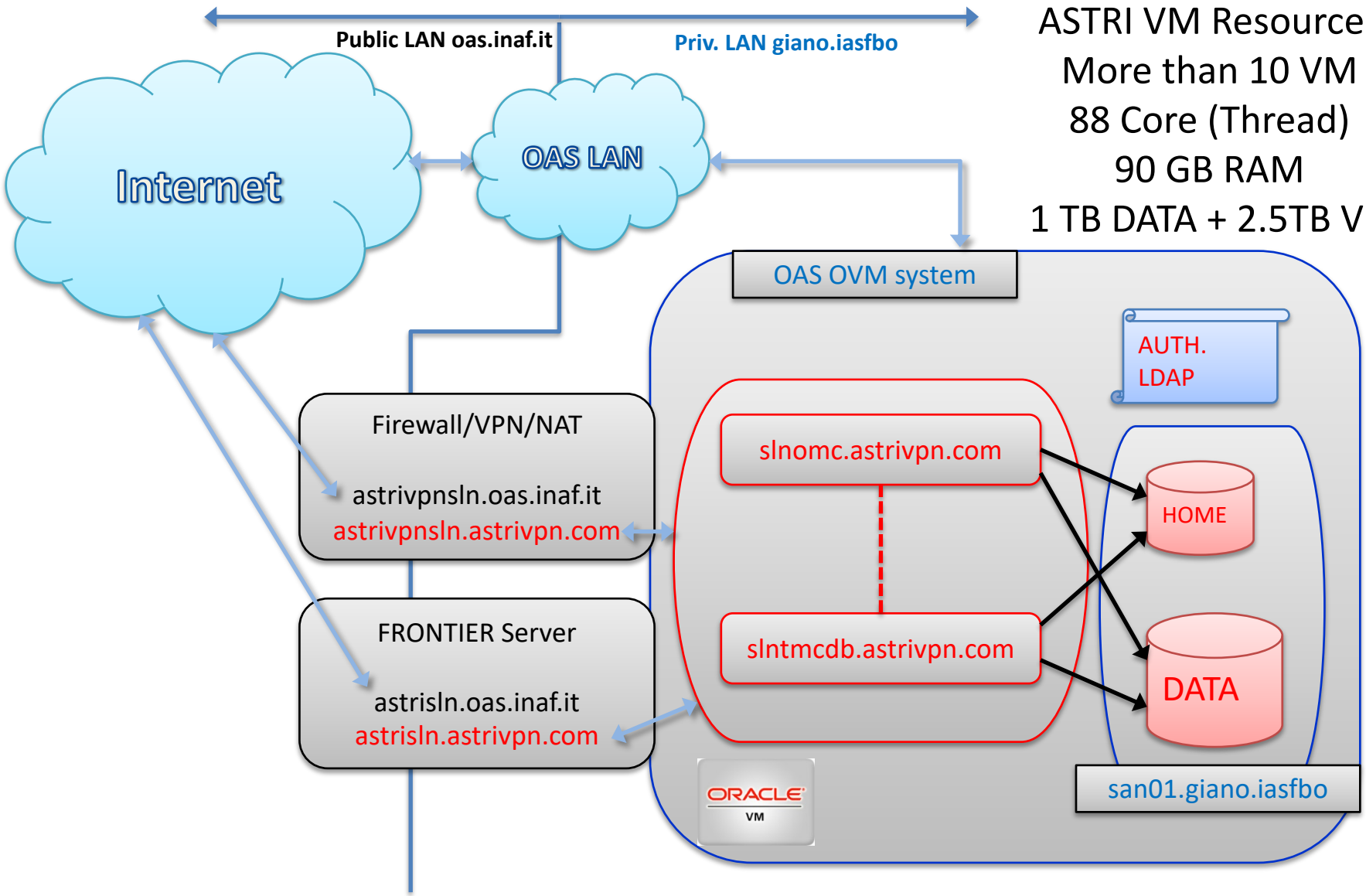
Public LAN 1Gbit (redundant)

SAN HA ISCSI
MULTI PATH
STORAGE

HW Resources
36 Phis. Core
72 Thread
384 GB RAM
8TB Storage

- 1 Oracle VM Manager
  Server SuperMicro X7DLV-E-B 2 Case 1U CPU Xeon 5140 64bit 2.33GHz 4MB 1333Mhz BUS dual core 8GB RAM 667MHz 4x2GB dual ch. HW RAID 2 HDD SATA II 300 GB RAID1 2x LAN 1Gbit. IPMI
- 3 Virtualization Servers
  DELL PowerEdge R620 Case 1U , Processors 2x6 core: Intel Xeon E5-2620v2 2.1GHz, 15M Cache, 7.2GT/s RAM 128GB. 4x1Gbit LAN. IDRAC7
- Storage Area Network System (SAN)
  DotHill 2U 12Bay 8 x iSCSI-1Gb to SAS 2GB Dual Controller System. 6 x HDD 2TB SAS 7.200 RPM in RAID 6 configuration. Redundant Control System.
- 2x24 1Gbit port switch

ASTRI Test Bed Architecture



ASTRI VM Resources
More than 10 VM
88 Core (Thread)
90 GB RAM
1 TB DATA + 2.5TB VM

The virtualization system is hosted in a private LAN of OAS (giano.iasfbo) that communicates with the Internet through the institute border switch.

- In the Test Bed they must be reproduced:
  - the same SLN servers, but virtual
  - the same network as the ASTRI prototype (192.168.100.0/24 astrivpn.com)
  - with the same IP addresses and server names as those of SLN.
- This is essential to make the SW run without changes

- External communication takes place via the Frontier server and the Firewall / VPN / NAT server. These will have different public IPs but names similar to those of the SLN prototype:
  - Frontier: (SLN) astrisln.oact.inaf.it => (TB) astrisln.oas.inaf.it
  - Firewall: (SLN) astrivpnsln.oact.inaf.it => (TB) astrivpnsln.oas.inaf.it
- The same network services must be guaranteed; DNS, VPN, NTP etc .. And also the same authentication system (LDAP / IPA)
- The giano.iasfbo network and astrivpn.com are not separated at level 2 (limit of the current virtualization system)

Storage space is essentially organized in two volumes:

HOME for user home and DATA for all other data.

- It is built in the SAN (san01.giano.iasfbo) which exposes LUN ISCSI multipath to the Frontier server, which then exports NFS volumes to all other servers.
- The same thing happens in the prototype where the Frontier Server exports its physical disks
- In the DATA volume the RAW and FITS scientific data are also stored contrary to what happens at SLN where there are dedicated NAS systems.

- This is a functional TB, we cannot do performance tests, we do not have sufficient computing power or sufficient storage space, nor is the storage fast enough.

- In TB all the physical devices are also missing, these are replaced by HW simulators so as to be able to integrate and test the Software in the TB in a complete and meaningful way.

TB was fundamental to support the upgrade of the Operating System (OS) from SL6.x to CentOS7.x

- First we created a new Test Bed TBA7 with CentOS7.x, solving the problems related to the OS change:
- Then we installed the ASTRI SW in TBA7
- Taking advantage of the TBA7 experience, we have reinstalled physical servers at SLN
- Now we are ready to install the ASTRI SW to SLN sure of not having big problems as everything has been installed in the TBA7
- The only problem was that it is not possible to run the old Test Bed TBA6 simultaneously with the new TBA7 due to OVM limitations

HW Limitations:

1. The servers are all over 5 years old and out of support. Especially the SAN, it is critical because it is difficult to find spare parts
2. The system does not guarantee a sufficient amount of CORE and RAM for the Mini Array TB.
3. The system has a limitation in storage performance because it is implemented by a SAN accessible via a 1GB network
4. Storage space is insufficient and cannot be easily increased => SAN with HD owners.

SW Limitations:

1. The OVM system presents the criticality of having a single control console (OVM Manager) that cannot be replicated.
2. Difficulty in accessing the VM console
3. Oracle VM is not secure enough when there is no system for taking VM snapshots or even a simple and / or automatic backup.
4. The Oracle VM SW, in the free version we use, is obsolete and does not allow the necessary upgrade to subsequent versions.
5. Oracle VM does not allow creating a virtual network structure suitable for simulating the Mini Array.

It is necessary to change both the HW and the SW.

The approach was to start from the search for a new Virtualization SW that would satisfy our needs and then understand the HW that we will need to create a sufficiently performing and reliable system.

We chose to adopt the SW **Proxmox VE**

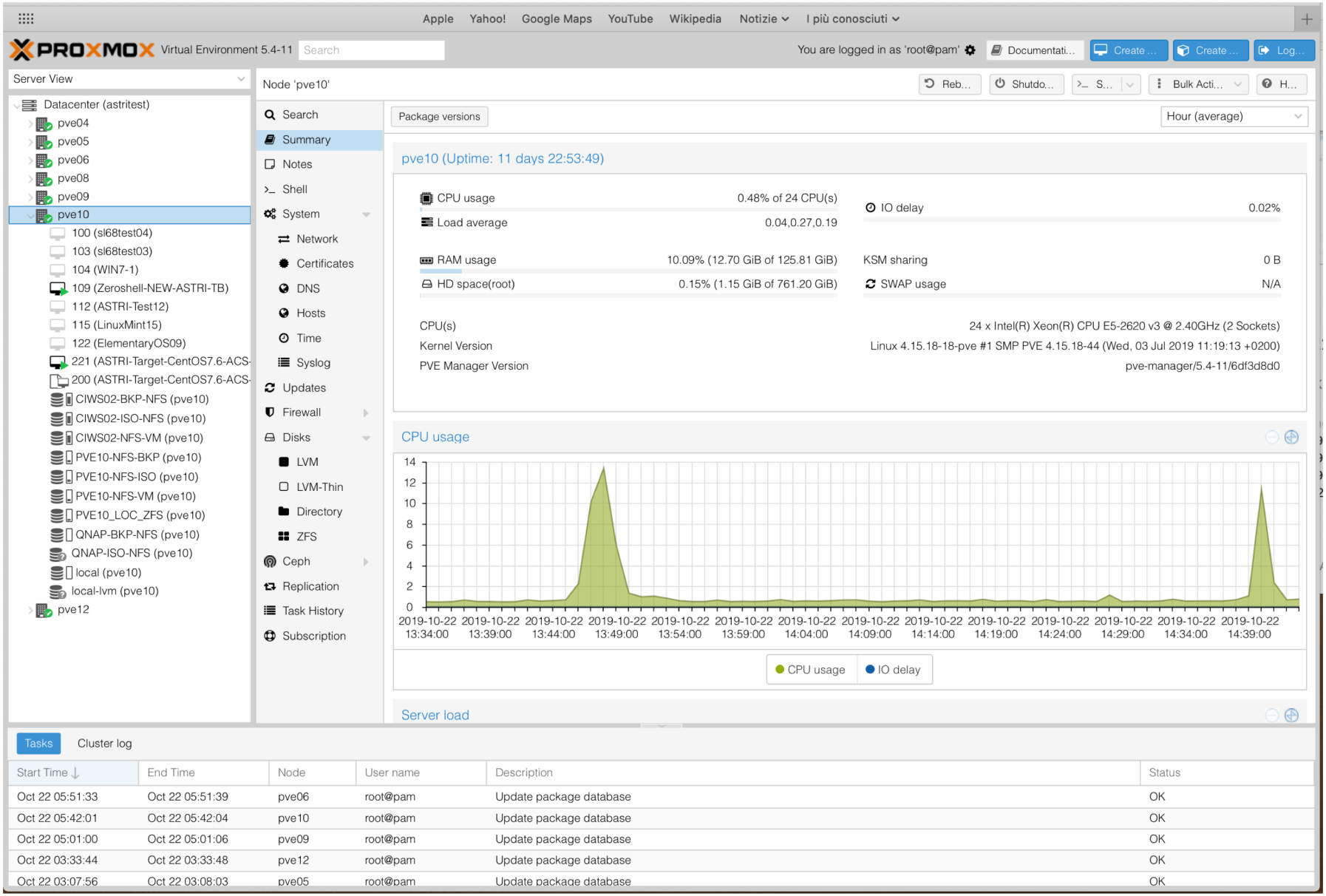
https://www.proxmox.com/

- Proxmox VE is a complete open-source platform for enterprise virtualization. With the built-in web interface you can easily manage VMs and containers, software-defined storage and networking, high-availability clustering, and multiple out-of-the-box tools on a single solution.

We did the first tests by creating a small teaching system that allowed us:

- to become familiar with Proxmox
- to touch the features we were looking for in the new virtualization system.

The choice of the ProxMox VE



We were able to verify and test that:

- It eliminates the criticality of the single control console because every Ipervisor can be used for this purpose.
- Allows easy upgrade to subsequent versions of the SW By freely accessing public repositories.
- It eliminates the need to resort to expensive SAN systems that are difficult to upgrade and maintain. In fact, storage can be achieved by organizing the IPs of the Ipervisors using the CEPH distributed file system.
- Provides the ability to make Snapshot and has a sophisticated VM backup system, both manual and automatic.
- Manages the virtual networks necessary for the Mini Array.

After the self-learning phase, we realized that: we needed a training course, that we organized in Julie 2019 the:

ProxMox VE Installation and and Administration

The course was held by SymTech IT s.r.l ProxMox VE Certified

- The course helped us to learn about the potential of ProxMox and to learn how:
- properly install an Ipervisor and create a virtualization cluster.
- install, import, modify and manage VMs in this new environment.
- Manage the high availability and VMs migration
- Create and manage distributed storage systems (CEPH)
- Create and manage virtual networks
- ……

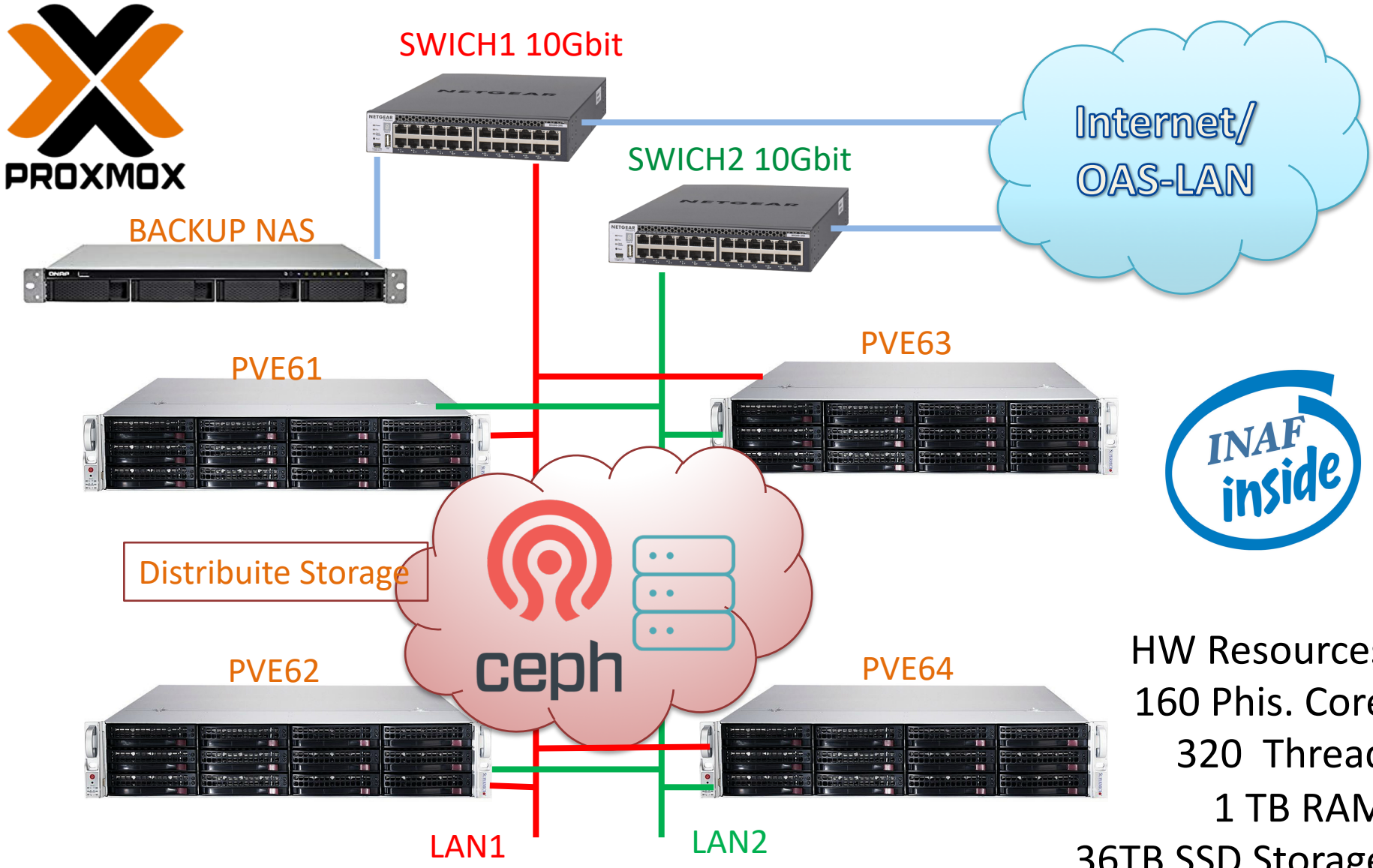
In the course we also learned:

- how to choose and properly dimension the HW to buy
- establish the HW costs more precisely.
- how to choose an appropriate SW license agreement, which is not too expensive.
- what we will have to do for the installation and commissioning of a professional system, even if we expect to avail ourselves of a consultancy from the same company that provided the course.

HW architecture of a ProxMox VE

- The HW simplification that come from the choice of ProxMox VE is remarkable in fact it can be eliminated:
- The Management Server
- The Storage Server (SAN)
- With only 4 Ipervisor servers and 2x10Gbit switches we are able to virtualize what will be needed for the ASTRI Mini Array at a cost comparable to that of the TB built for ASTRI Prototype.
- However in the final project we have chosen to also add an economic NAS for VM and Data backup
- HW simplification with increased and improved functionality and performance!

HW architecture of a ProxMox VE



HW Resources
160 Phis. Core
320 Thread
1 TB RAM
36TB SSD Storage

- 4 x ProxMox VE Ipervisor Servers Supermicro Motherboard and 2U Case with 1000Watt Redundant Power, Processors 2x Intel Xeon 20-Core 6230 2,1Ghz 27.5MB Cache 10.4GT/s, RAM 256GB, 2x10Gbit RJ45 LAN, IPMI, Storage HBA with 2x240GB SSD + 6x2TB SSD
- NAS QNAP 1U redundant Power, 4 Core AMD Processor, 8GB RAM, 4 x 6TB HDD SATA HD in RAID5, 4x1Gbit LAN + 10Gbit RJ45 LAN.
- 2x24 10Gbit RJ45 port managed switch. Netgear M4300-24x

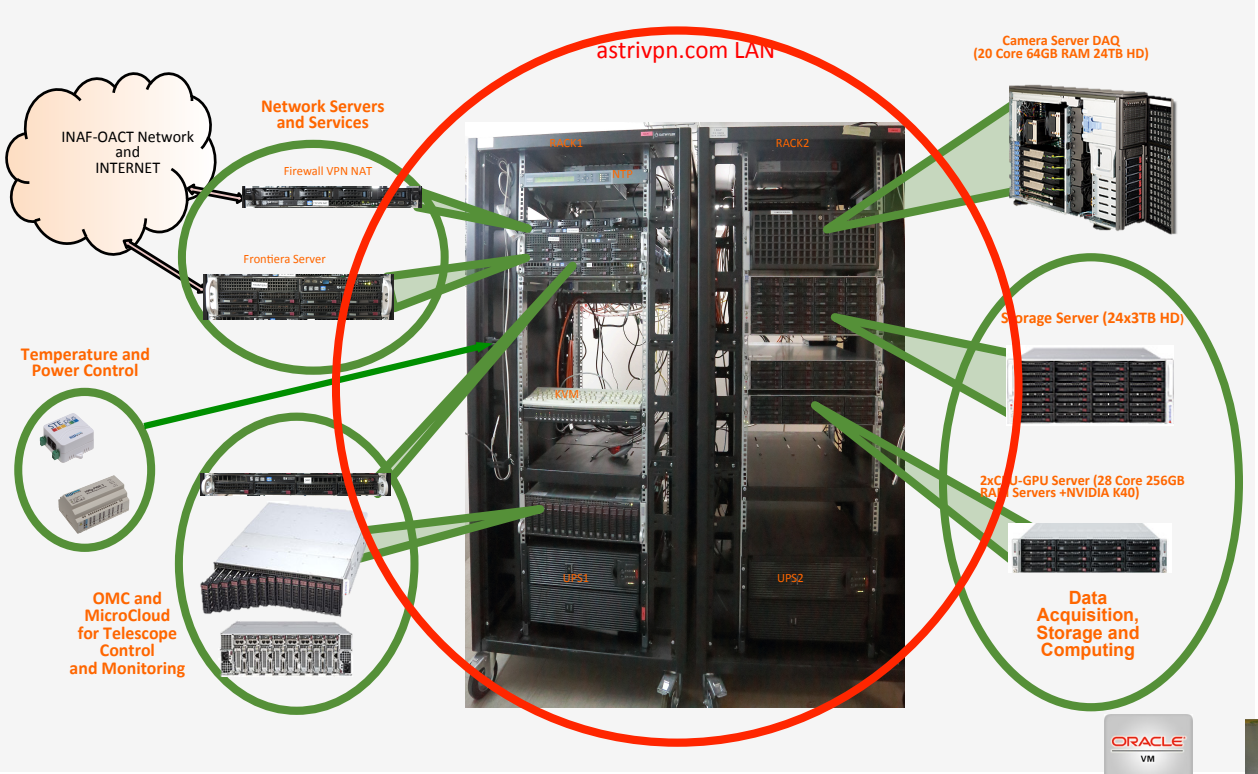
CONCLUSIONS and Developement

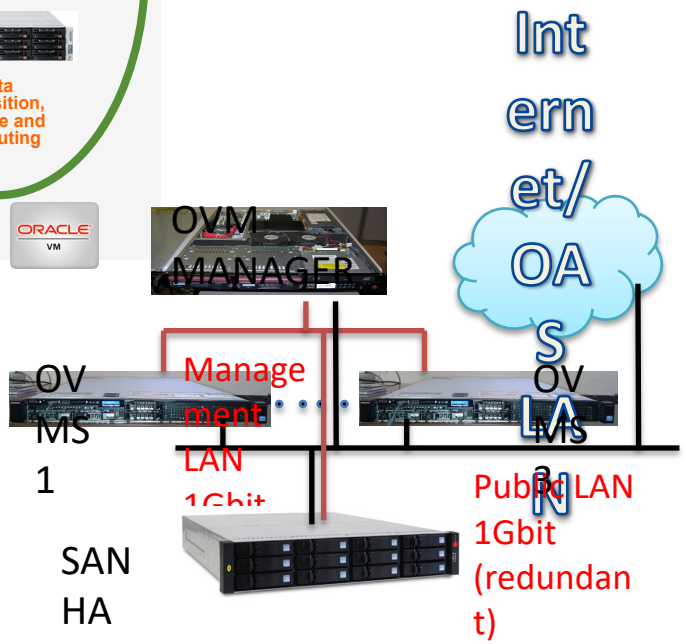
- We found a virtualization SW, ProxMox, which solves all the limitations we have with the current Oracle VM System.
- We are already experimenting on how to bring TB ASTRI to ProxMox and we plan to start soon the first installation tests of the ASTRI SW on this installation.
- We have identified the HW infrastructure of the new virtualization system.
- Now we must implement it and put it into operation by the first half of 2020
- In the meantime we will have to design the TB architecture of ASTRI's Mini Array to start implementing it in this infrastructure.
- We will have to check if: **The virtual approach used for the Test Bed can be used for the telescope control and monitor of the ASTRI Mini Array**

Thank you for your attention!

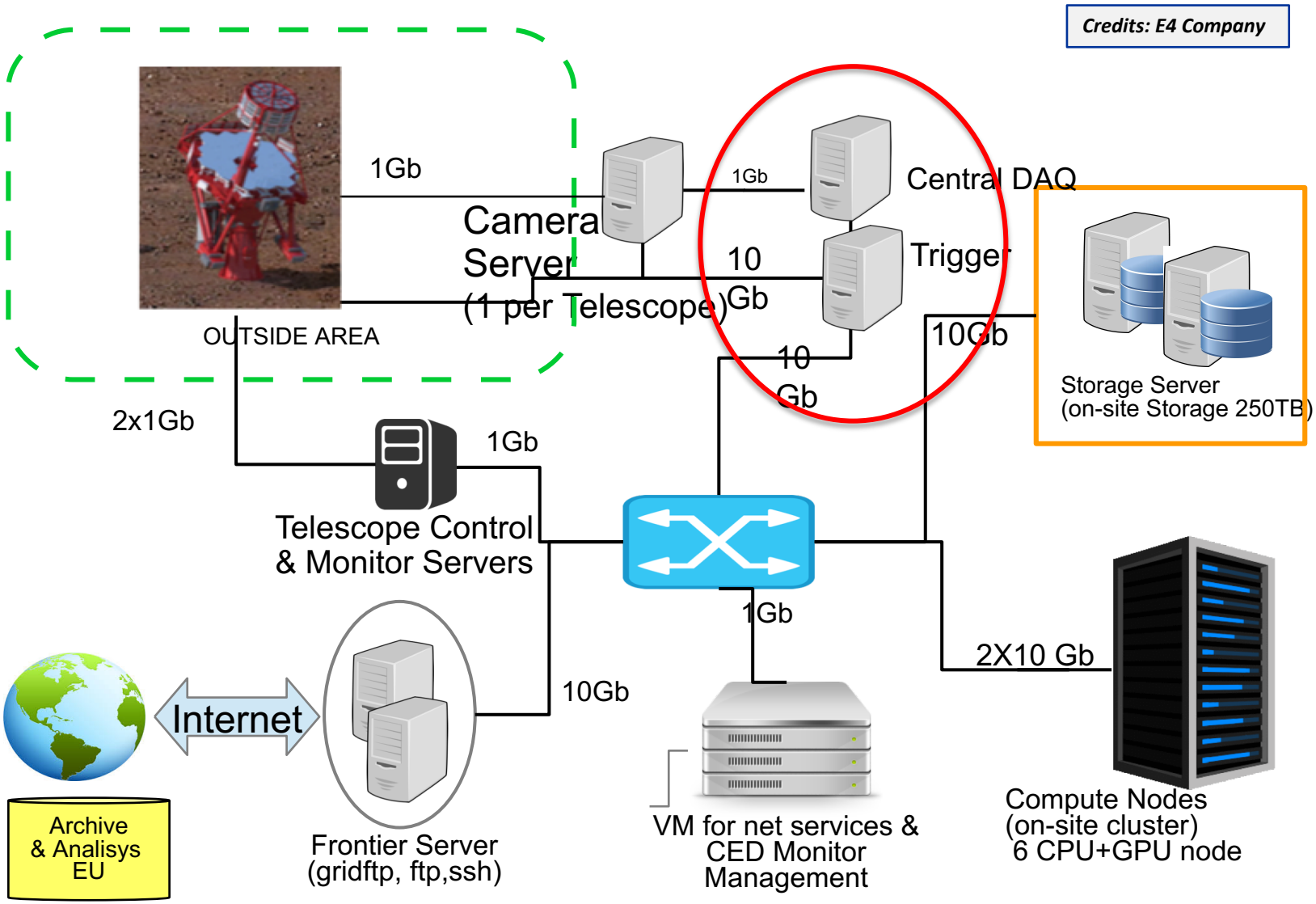
Spare SLIDES

ASTRI Test Bed for SW Development





Mini Array Computing Schema



Principali differenze fra L'ICT ASTRI e MiniArray

- Aggiunta del Central DAQ
- Aggiunta del Central Trigger
- Evoluzione dello storage dal sistema composto da File Server, server con molti HD, a quello di Filesystem parallelo e distribuito come LUSTRE (http://lustre.org) o meglio BeeGFS (https://www.beegfs.io)
- LUSTRE è lo stesso sistema adottato al sito Nord di CTA da LST. BeeGFS è il suo successore.
- Virtualizzazione dei server dedicati alla movimentazione dell'ARRAY. La soluzione virtuale è quella scelta per tutti i futuri progetti simili a ASTRI CTA. Cosa constatato allo SPIE 2018

Server categories

1. Camera Server
2. Central DAQ
3. Trigger (Timing)
4. Observatory, Telescope and Camera Control and monitoring
5. On-Site Analysis
6. On-Site Storage
7. Service Servers

For each of these categories we will have to determine the number of required servers and their characteristics
With the help of the groups working on the related SW

Server main caracteristics

- CPU, CPU Core
- RAM
- HD and HD Capacity
- RAID and RAID Level
- Network type and Number of Interface

After this:

- Choosing the server model respecting the desired performance the necessary reliability
- Trying to standardize as much as possible servers in a few different models

Network main caracteristics

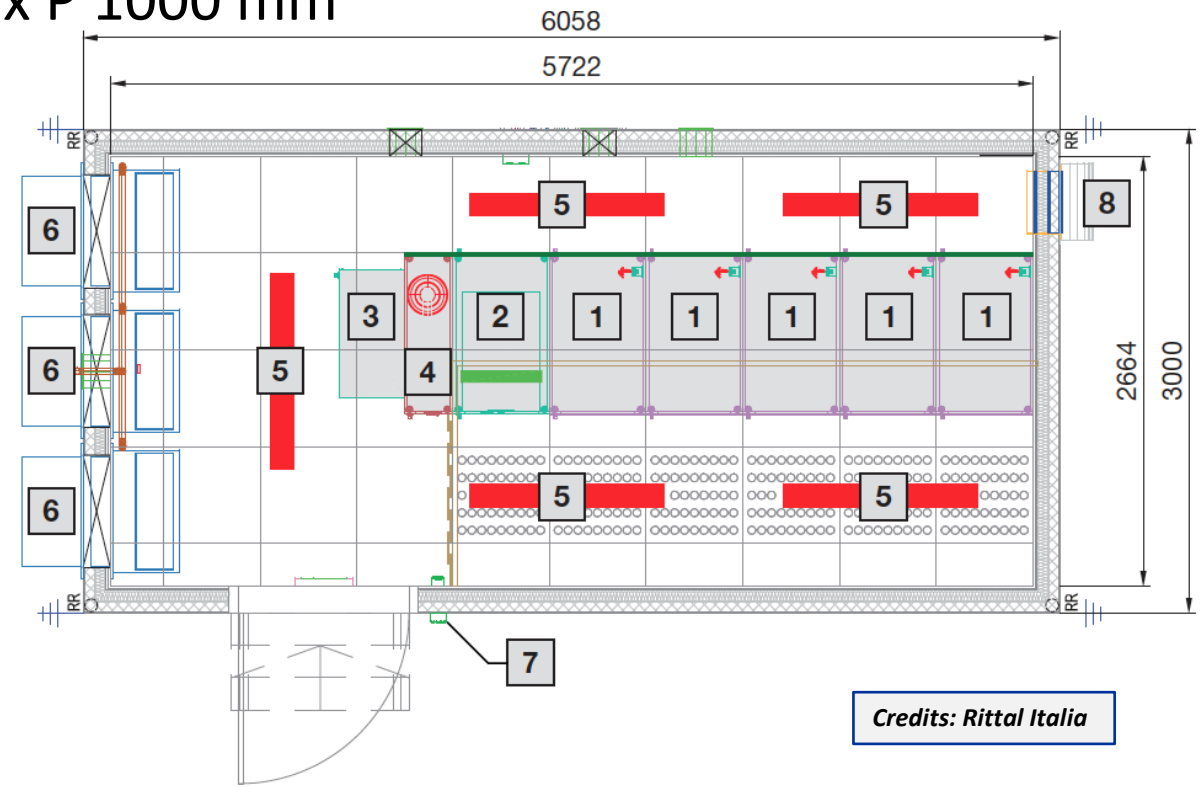
- LAN
  - Realized by 1-10Gbit/s Ethernet switch
  - Dimensioned according to the number of expected servers
- Telescope to Server Room connection
  - Fiber Optics LAN connection -> we will refer to the documents produced by CTA
- Internet we should have:
  - 2-4 public IPv4 addresses with global connection speeds of 100Mbit / s. For VPN, NAT and frontiera Servers
  - 1-2 public IPv4 addresses with global connection speeds of 1Gbit / s for data transfer.
- Timing LAN: low-latency network for the time and the trigger distribution
  - 1Gbit/s Dedicated switch (white rabbit?)

Data Center in a BOX

Assembled, installed and tested at home, just connect the data network and the power and turn on

Container -> L 6050-8000 x A 3250 x P 3000

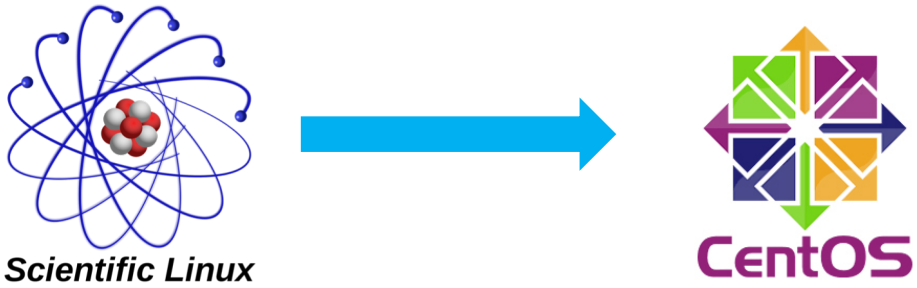
1. Rack L 600 x A 2000 x P 1000 mm
2. UPS Rack
3. Electrical board
4. Fire system
5. Lighting
6. Cooling System
7. Access control
8. Air Pressure Control



Credits: Rittal Italia

10 ASTRI Telescope Case

- 10xCamera Server (16Core 32GB 4TB) 5.5KE
- 3xCentral DAQ (24Core 256GB 44TB) 11,3KE
- 2xTrigger (Timing) (24Core 256GB 4TB) 11,3KE
- 8xObservatory, Telescope and Camera Control and monitoring (16Core 32GB 4TB) 5.9KE
- 6xOn-Site Analysis (24Core 256GB 4TB 1GPU) 20KE
- 4x On-Site Storage (250TB available)
- 3x Service Servers + 1 SAN (20TB)
- 2X Frontier Server (16Core 32GB 4TB) 5.9KE
- 4x10Gbit + 4x1Gbit network switch
- 4 x 42 Unit computer rack
- 8x 10KW/h UPS
- Electrical power 35KW (TBV) excluding Cooling
- Everything can fit in a Data Center Container (see slide 5)
- Cost -> less than 700KE tax included excluding container



Changing Operating System from SL6.x to CentOS7.x

- The transition to the CentOS7.x Operating System is mainly necessary to run the latest versions of ACS
- The main problem with this operation is that it is not an Upgrade, but a Reinstallation
- The idea was to:
  - start developing the software first in a New Virtual Machine with CentOS7.x and the latest version of ACS
  - integrate it using a New Virtual Test Bed with CentOS7.x
  - and finally install SW at SLN where all the servers will need to be reinstalled with CentOS7.x

- Creation of the new development virtual machine with CentOS7.4 and ACS JUN2017
  ftp://astrisw_ftp@astri.iasfbo.inaf.it/VMS/ASTRI_Target_CentOS7.4_ACS-JUN2017_OCT2017.ova

- Creation of a new test bed with CentOS7.x (TBA7):
  - we have imported the development VM into the TB replicating it and configuring it for the characteristics of each server needed in the testbed.
  - we had to adopt and install a new FreeIPA authentication system because the old OpenLDAP was not compatible with CentOS7.x and also importing the user database.
  - the new TBA7 and the old Test Bed (TBA6) can not run together and this complicates the work.
  - we are almost ready to start the first tests of the Software

- CentOS7.x installation on SLN servers was made using a dedicated hard disk set.
- Each server has two sets of disks with one with the SL6.x installed and another dedicated to installing CentOS7.x
- At the moment we have completed the installation of CentOS7.x for all servers, but the system will remain with the SL6.x until we are ready with the new Software.
- It is essential to carefully prepare the switch phase between SL6.x and CentOS7.x because it is not as simple as for the Test Bed to switch from one to another Operating System.