ITMO UNIVERSITY

Information Technology, Mechanics, and Optics
St.Petersburg, Russian Federation

Cloud platform for Geographically Distributed Data Centers

Contract # 03.G25.31.0229 “Development of new technological components for management systems of geographically distributed Data Centers”

GRID2018, JINR, Dubna, 14, September, 2018, Reporter: Andrey Shevel

Outline

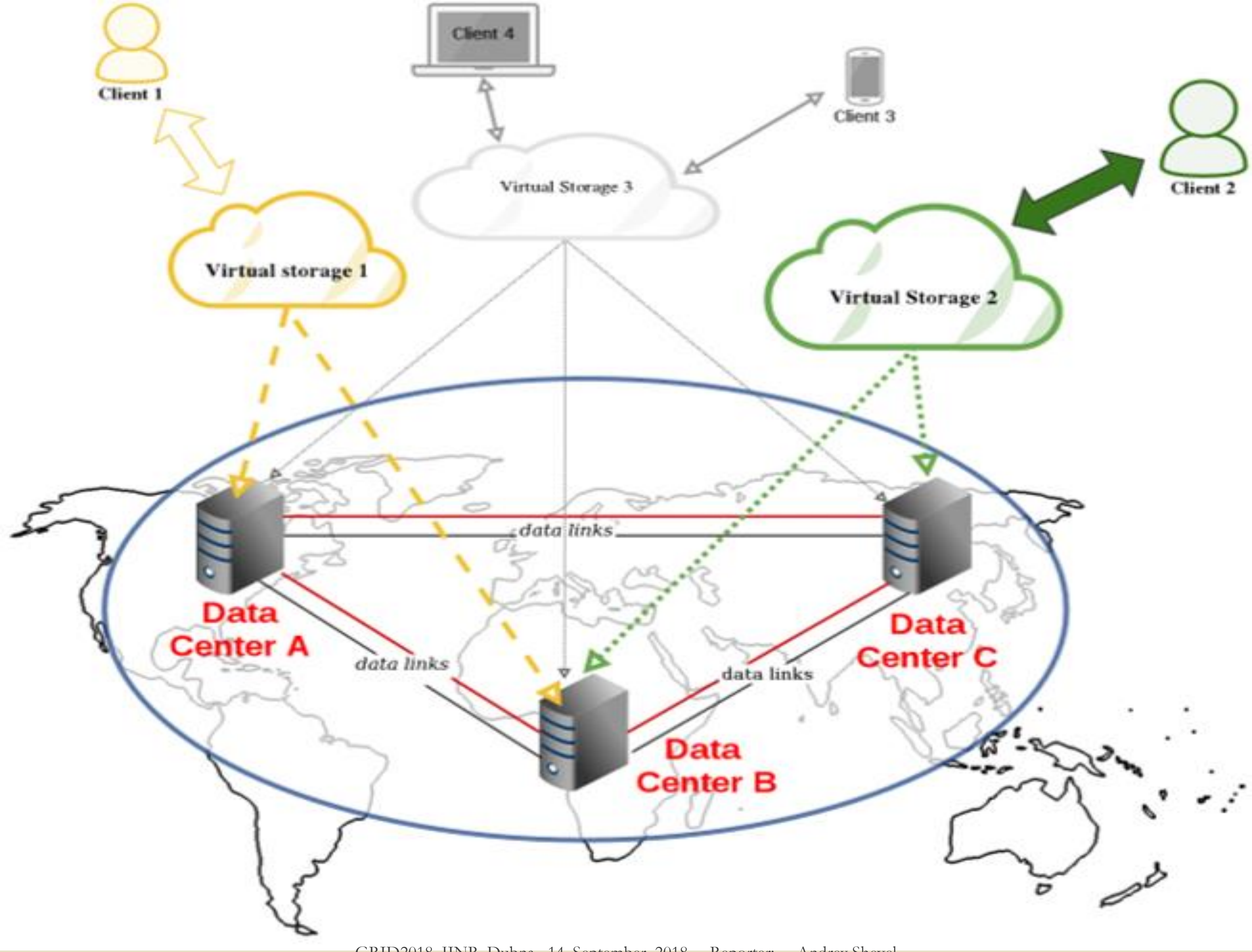
- Clouds and distributed computing tendency
- Current project
- Requirements and Solutions
- Data transfer, Storage, and Computing
- Authentication and Security
- Conclusion

Contemporary computing tendencies

- Number of Data Centers in the World is steadily growing.
- A lot of efforts to move from concrete DCs to public clouds are taken place. HEP is not exclusion.
    - There are many tests/attempts in HEP to use public clouds.
        - Tony Wong "The Role of Dedicated Data Computing Centers in the Age of Cloud Computing" (CHEP2016).
        - John Hover "Running ATLAS at scale on Amazon" (HEPIX-spring-2015)
- Moving to multi cloud and multi DCs is obvious.

Multi DCs use tendency

- Demands to integrate resources of many DCs are growing.
  - Pros:
    - Ability to redistribute resources in according to changing demands.
    - More reliable service even in case when one DC is down.
    - Using one even large public cluster has drawbacks
      - Lack of isolation.
      - Lack of elasticity.
  - Cons:
    - It is needed special management system to control multiple DCs;
    - Special attention to data links between DCs is required.
- Current project is aimed to bring the Scalable Geographically Distributed DCs conception into reality.


Client 1
Client 4
Client 3
Client 2
Virtual Storage 3
Virtual storage 1
Virtual Storage 2
data links
Data Center A
data links
data links
Data Center C
Data Center B

Main intentions for Scalable Geographically Distributed DCs

- Assume all DCs are “dark” DCs
- Develop integrated control system/center
- Subscriptions for registered users/organizations:
  - Use public cloud of type IaaS.
  - Create and use private cloud of type IaaS.

Approaches

- Software Defined Storage (SDS)
- Software Defined Network (SDN)
    - Network Function Virtualization (NFV).
- Infrastructure as Code (IaC)

Technical Requirements for system management of Scalable Geographically Distributed DCs

- Data links
  - Hardware security.
  - Data compression/decompression.
  - Data Encryption/decryption.
- Service reliability
  - Steady system functioning.
  - Gradual degradation with hardware or software malfunction.
- Monitoring physical and virtual infrastructure.
- Data storage features include the long data storage with total volume until 1 EB.
- SLA support.
- Data Center Infrastructure Management (DCIM).
- Automatic deployment of the system management (separate presentation P. Fedchenkov "*Approaches to the automated deployment of the cloud infrastructure for geographically distributed data centers*").

Architecture technical solutions

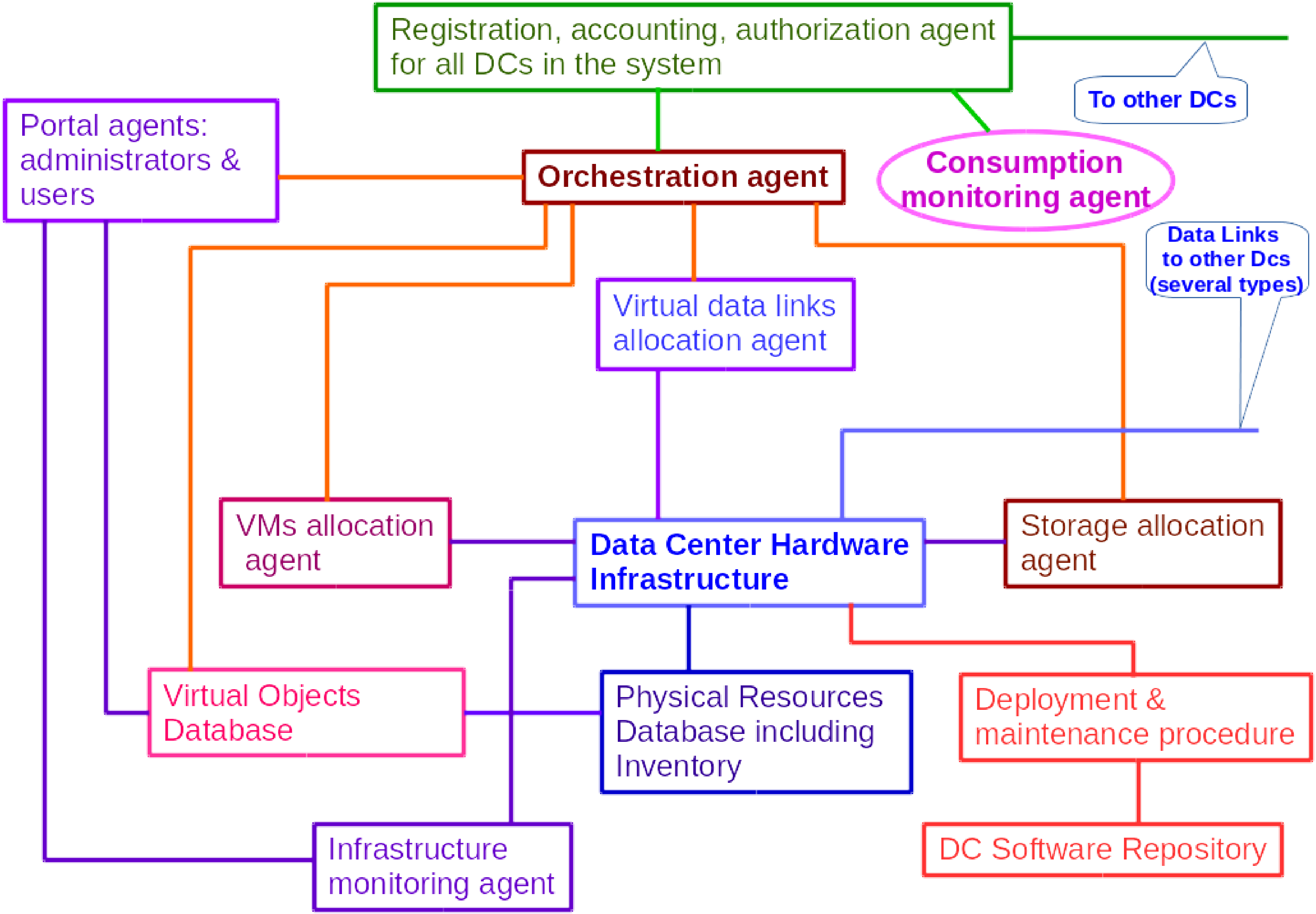
- System Management Architecture of Scalable Geographically Distributed DCs
  - Implemented in form of loosely coupled program agents.
  - The system is based on FOSS components.
- Operating System - Naulinux (compatible with RedHat, Scientific
- Linux, CentOS).
- Computing and portals – Openstack.
- Data Storage - Software Defined Storage (SDS) – CEPH.
- Monitoring of physical and virtual infrastructure – Zabbix.
- Visualization – Grafana/Kibana.
- Communication security - Symmetric cryptography (OpenSSL) with
  - Quantum Key Distribution (QKD) procedure.
- Independence of external changes (versions, polices, etc) – own versions of
  - Naulinux distribution + own code **repository**.
- Each type of service has to be feasible for accounting purposes.

Explanations

- **Agent** in the management system is
  - a program which is running like daemon in isolated operating environment from the start of the system;
  - **agent** accepts requests from other program or another **agent** in according to developed protocol over a range of queues to increase service reliability and horizontal scalability.
  - If any agent is down it does not mean the management system is down.
- **Naulinux** is Linux distribution developed ~10 years ago with support from Education Ministry of Russian Federation. The developed rpm repository permits to be more independent from external changes like software versions, software policy availability, etc.
- **Portals** are based on Openstack horizon infrastructure.

DC interconnection architecture

- All DCs are connected over control private network with OpenVPN:
  - each server has dedicated network port to be connected to mentioned network.
- All DCs have the replica of system control directory entitled **Registry.**
  - The Registry does keep all required parameters for the system (databases locations, data links between DCs, etc)
  - Each agent has to read regularly the content of **Registry**.



Registration, accounting, authorization agent for all DCs in the system

To other DCs

Portal agents: administrators & users

Orchestration agent

Consumption monitoring agent

Data Links to other Dcs (several types)

Virtual data links allocation agent

VMs allocation agent

Data Center Hardware Infrastructure

Storage allocation agent

Virtual Objects Database

Physical Resources Database including Inventory

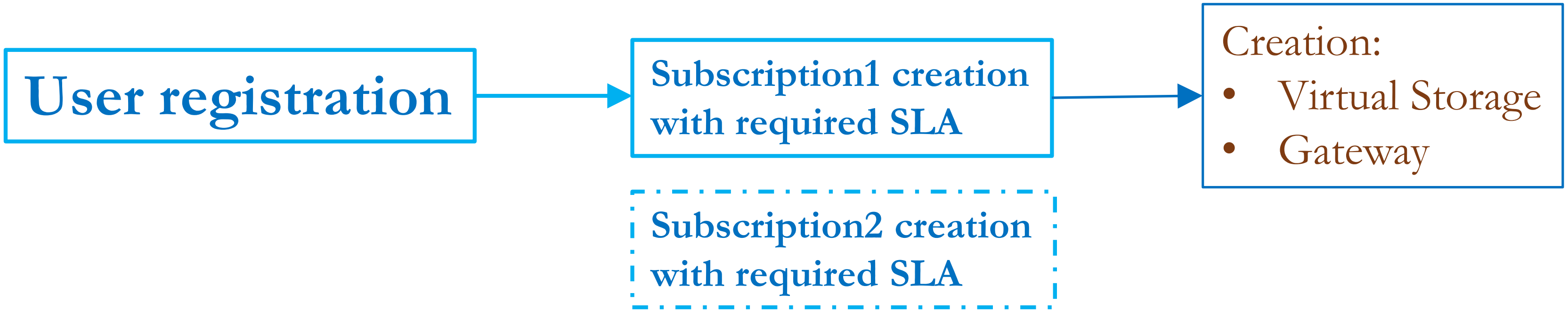
Deployment & maintenance procedure

Infrastructure monitoring agent

DC Software Repository

SLA implementation for data storage

- SLA for data storage is mainly implemented by CEPH
- The list of possible SLA for Virtual Storage might include:
  - Object storage (over gateway)
  - File storage (over gateway with or without Nextcloud)
  - Block storage (over gateway)
  - Number of replicas
  - Size of Virtual Storage
  - Erasure coding
  - Data Encryption
  - Data Compression



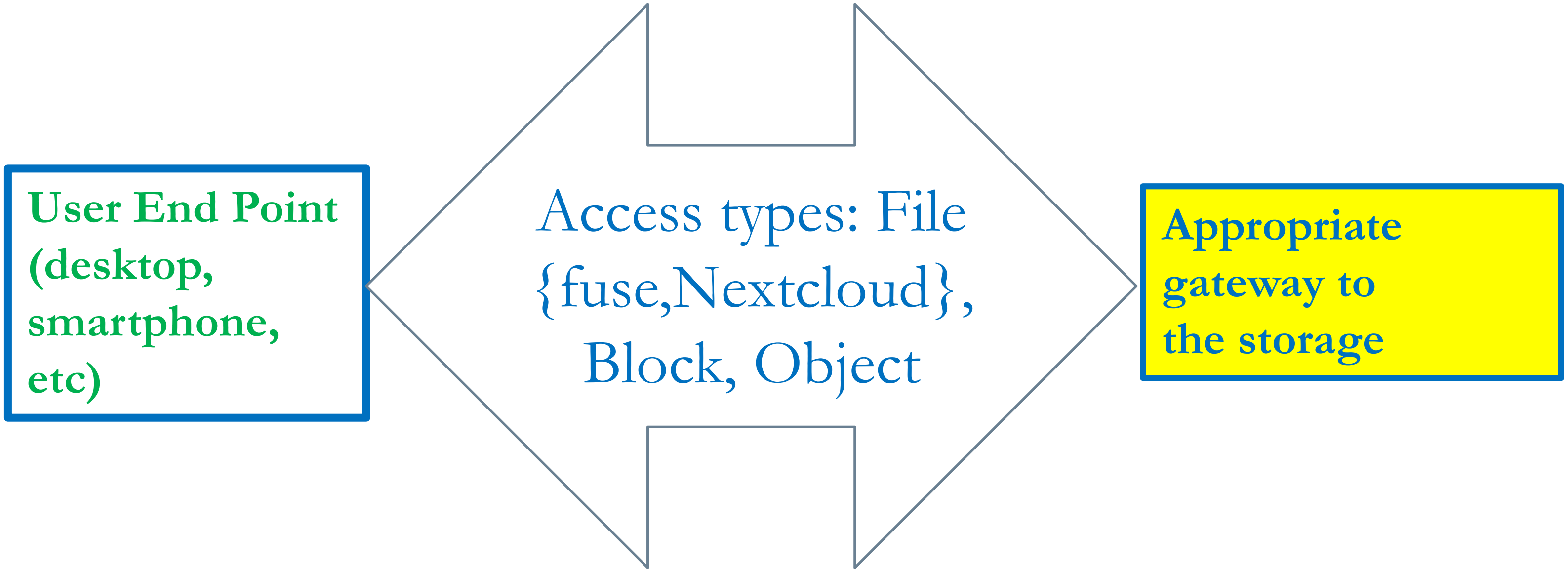
User registration

Subscription1 creation with required SLA

Subscription2 creation with required SLA

Creation:

- Virtual Storage
- Gateway



User End Point (desktop, smartphone, etc)

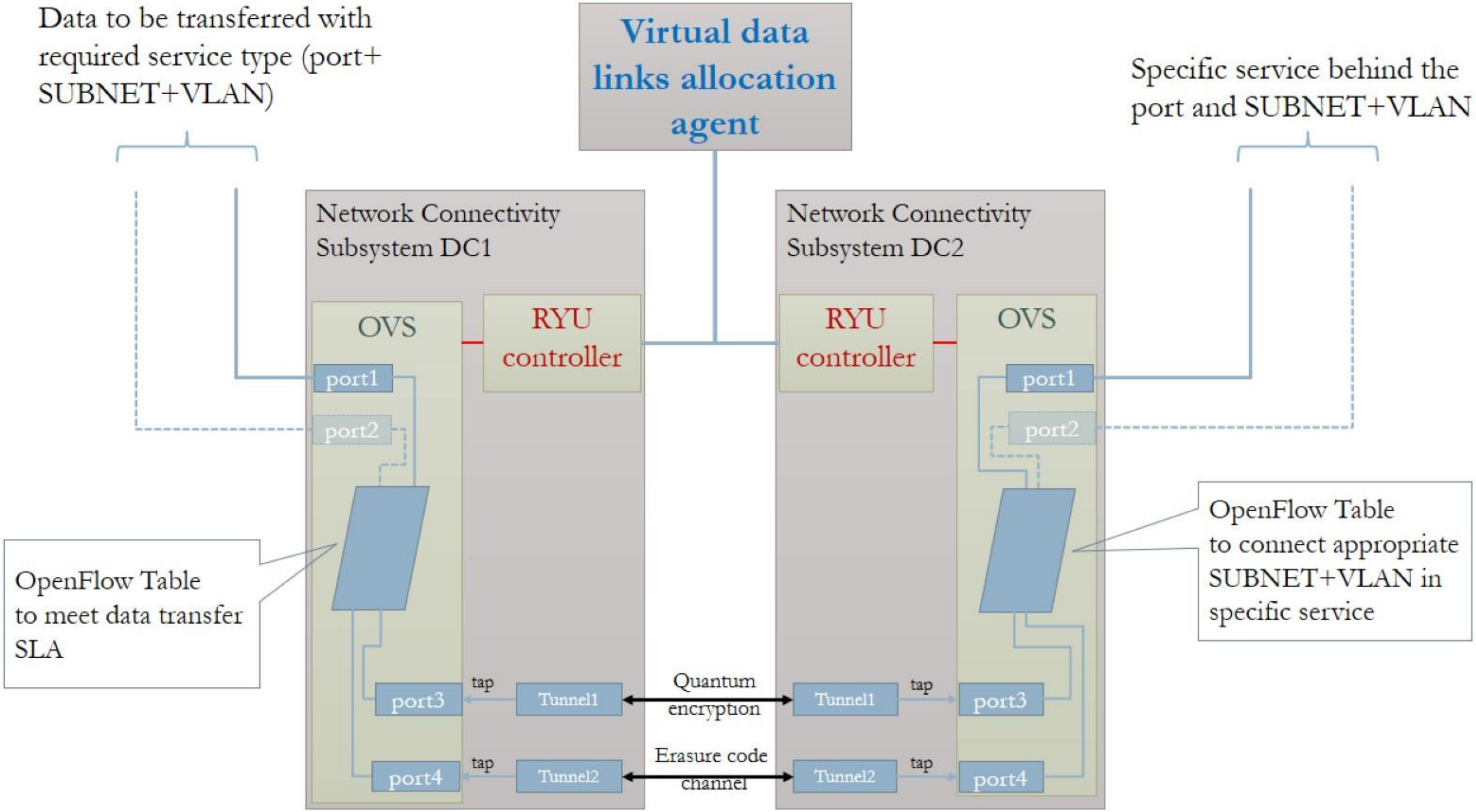
Access types: File {fuse,Nextcloud}, Block, Object

Appropriate gateway to the storage

SLA implementation for data transfer

- To meet SLA the system management uses specially configured network architecture where each kind of concrete SLA is mapped to concrete subnet segment. VMs connected to above concrete subnet segment can send and receive the data with above concrete SLA.
- The list of SLA for data transfer might include:
  - Data compression / decompression
  - Data encryption / decryption
  - Erasure coding / decoding

Data transfer implementation with SDN Openflow


Data to be transferred with required service type (port+ SUBNET+VLAN)
Virtual data links allocation agent
Specific service behind the port and SUBNET+VLAN
Network Connectivity Subsystem DC1
Network Connectivity Subsystem DC2
OVS
RYU controller
RYU controller
OVS
port1
port2
port1
port2
OpenFlow Table to meet data transfer SLA
OpenFlow Table to connect appropriate SUBNET+VLAN in specific service
port3
tap
Tunnel1
Quantum encryption
Tunnel1
tap
port3
port4
tap
Tunnel2
Erasure code channel
Tunnel2
tap
port4

VM management

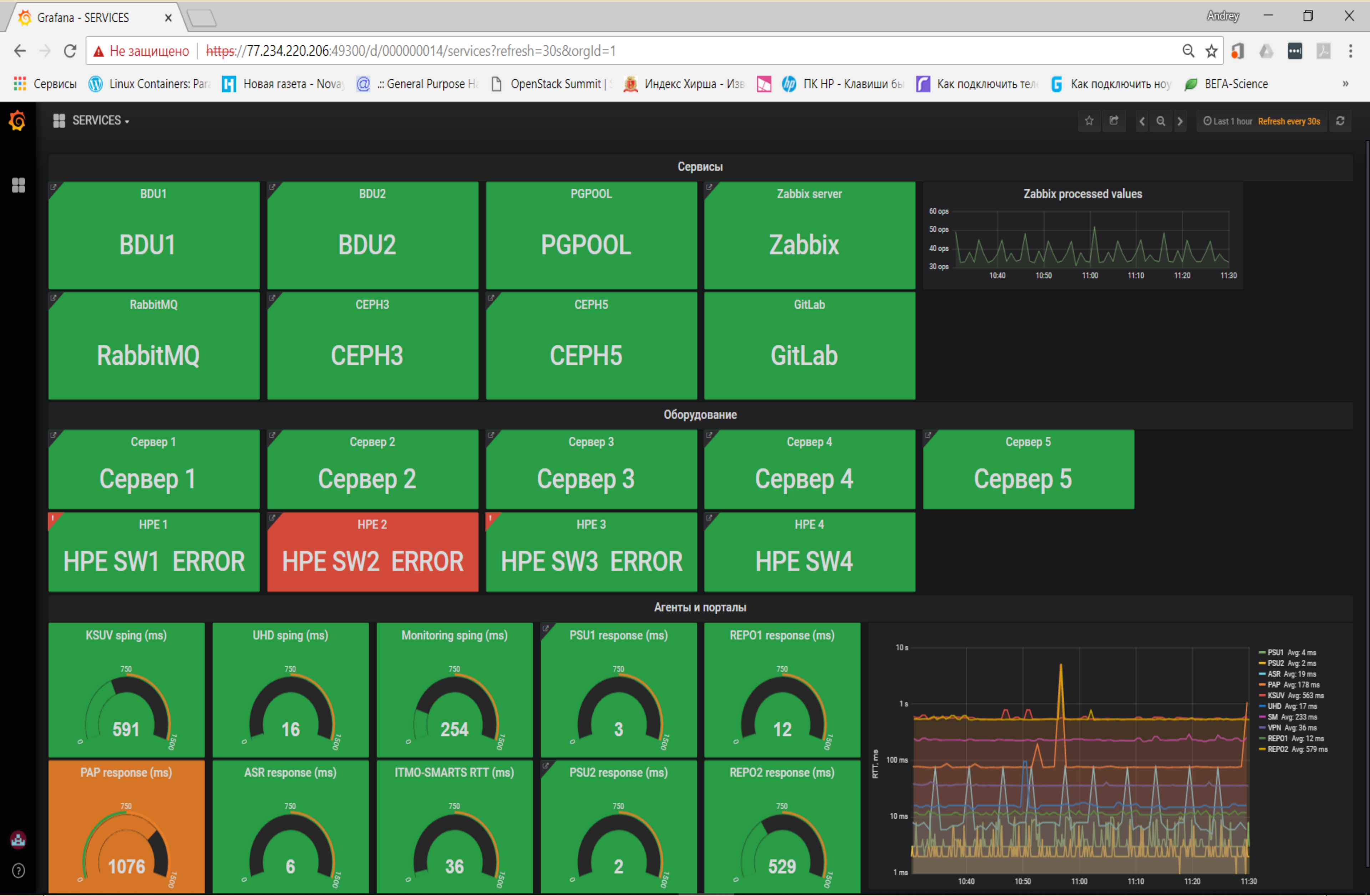
- VMs are created and kept by Openstack.
- VM features in the project:
  - VMs might be created in different DCs.
  - VMs user/owner might have reserved a number of storage types.
  - User might create desired architecture , e.g. Grid or something different.

Authentication management

- All registration procedures and subscriptions creation are performed in separate accounting system which is out of the presentation scope.
- Inside the system management all users are identified by UUID generated in accounting system (no any connection with real user names).
- All networks inside above system are implemented with OpenVPN.

Current test configuration

- Several DCs: servers [6 + 2 + 2]: HPE DL380 Gen10 8LFF CTO
  - CPU : 2 x Intel Zeon Gold 6130
  - Main memory: 128 GB
  - Disk storage 32 TB
- Several switches: HP FF 5700-32XGT-8XG-2QSFP+ Switch


Grafana - SERVICES
Andrey
Не защищено
https://77.234.220.206:49300/d/000000014/services?refresh=30s&orgId=1
Сервисы
Linux Containers: Par
Новая газета - Nova
.: General Purpose H
OpenStack Summit |
Индекс Хирша - Изв
ПК HP - Клавиши бы
Как подключить тел
Как подключить ноу
ВЕГА-Science
SERVICES
Last 1 hour Refresh every 30s
Сервисы
BDU1
BDU1
BDU2
BDU2
PGPOOL
PGPOOL
Zabbix server
Zabbix
Zabbix processed values
60 ops
50 ops
40 ops
30 ops
10:40
10:50
11:00
11:10
11:20
11:30
RabbitMQ
RabbitMQ
CEPH3
CEPH3
CEPH5
CEPH5
GitLab
GitLab
Оборудование
Сервер 1
Сервер 1
Сервер 2
Сервер 2
Сервер 3
Сервер 3
Сервер 4
Сервер 4
Сервер 5
Сервер 5
HPE 1
HPE SW1 ERROR
HPE 2
HPE SW2 ERROR
HPE 3
HPE SW3 ERROR
HPE 4
HPE SW4
Агенты и порталы
KSUV sping (ms)
591
UHD sping (ms)
16
Monitoring sping (ms)
254
PSU1 response (ms)
3
REPO1 response (ms)
12
PAP response (ms)
1076
ASR response (ms)
6
ITMO-SMARTS RTT (ms)
36
PSU2 response (ms)
2
REPO2 response (ms)
529
0
750
1500
10 s
1 s
100 ms
10 ms
1 ms
RTT, ms
10:40
10:50
11:00
11:10
11:20
11:30
PSU1 Avg: 4 ms
PSU2 Avg: 2 ms
ASR Avg: 19 ms
PAP Avg: 178 ms
KSUV Avg: 563 ms
UHD Avg: 17 ms
SM Avg: 233 ms
VPN Avg: 36 ms
REPO1 Avg: 12 ms
REPO2 Avg: 579 ms

Running Project Advantages

- Management software based on loosely coupled software agents in form of VM or containers:
  - no conflicts with versions/components requirements in different agents;
  - an agent down does not mean whole system is down.
- Data links between DCs might use QKD procedure.
- Own repository (binary and source codes).

Conclusion

- The Scalable Geographically Distributed DCs prototype is in intensive debugging and moving to production stage.
- Experimental distributed storage is planned to be available until end of September 2018.
- Main tasks to be developed/performed:
  - procedure to determine a degradation level when an infrastructure component is down;
  - monitoring subsystem;
  - automatic system deployment.
- Volunteers who are interested in the development participation are welcome.

References

- Cloud platforms
  - http://openstack.org
  - https://dcos.io/
  - http://cloudstack.apache.org/
  - https://www.eucalyptus.cloud/index.html
  - https://opennebula.org/
- Monitoring & visualization
  - http://zabbix.org/wiki/Main_Page
  - https://grafana.com/
  - https://www.elastic.co/products/kibana
- NFV
  - https://www.opnfv.org/
- IaC
  - https://saltstack.com/community/
  - https://puppet.com/
  - https://www.chef.io/
  - https://www.ansible.com/
  - https://www.terraform.io/
- Initial deployment
  - https://xcat.org/
  - https://docs.openstack.org/ironic/latest/

Thanks for your attention.
Now is time for questions.