OpenStack: Security beyond firewalls

Giuseppe “Gippa” Paternò, Network & Security NERD
30th May 2014 * OpenStackDay Italy

Twitter: @gpaterno - Website: www.gpaterno.com

GARL
MAKING THE CLOUD A SAFER SPACE

About me

IT Architect and Security Expert with background in **Open Source**. Former Network and Security architect for Canonical, RedHat, Wind/ Infostrada, Sun Microsystems and IBM and Visiting Researcher at the University of Dublin Trinity College.

Past projects: standard for J2ME Over-The-Air (OTA) provisioning along with Vodafone, the study of architecture and standards for the delivery of MHP applications for the digital terrestrial television (DTT) on behalf of DTT Lab (Telecom Italia/LA7) and implementation of HLR for Vodafone landline services.

Lot of writings, mainly on computer security.

CTO and Director of **GARL**, a multinational company based in Switzerland and UK, owner of SecurePass and SecureData.

GARL

IT security products and virtualization services focused on identity protection on the **Cloud**, as the user is became the ultimate perimeter of a never ending distributed model.
HQ based in **Switzerland** and whose servers are located in Switzerland.
User privacy is protected by strict Swiss privacy regulations, no UE or US exceptions allowed.

Too many threats

62%

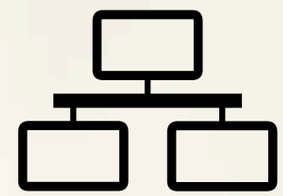
Increase breaches in 2013[1]

1 in 5

Organizations have experienced an APT attack [4]

3 Trillion$

Total global impact of cybercrime[3]



2,5 billion

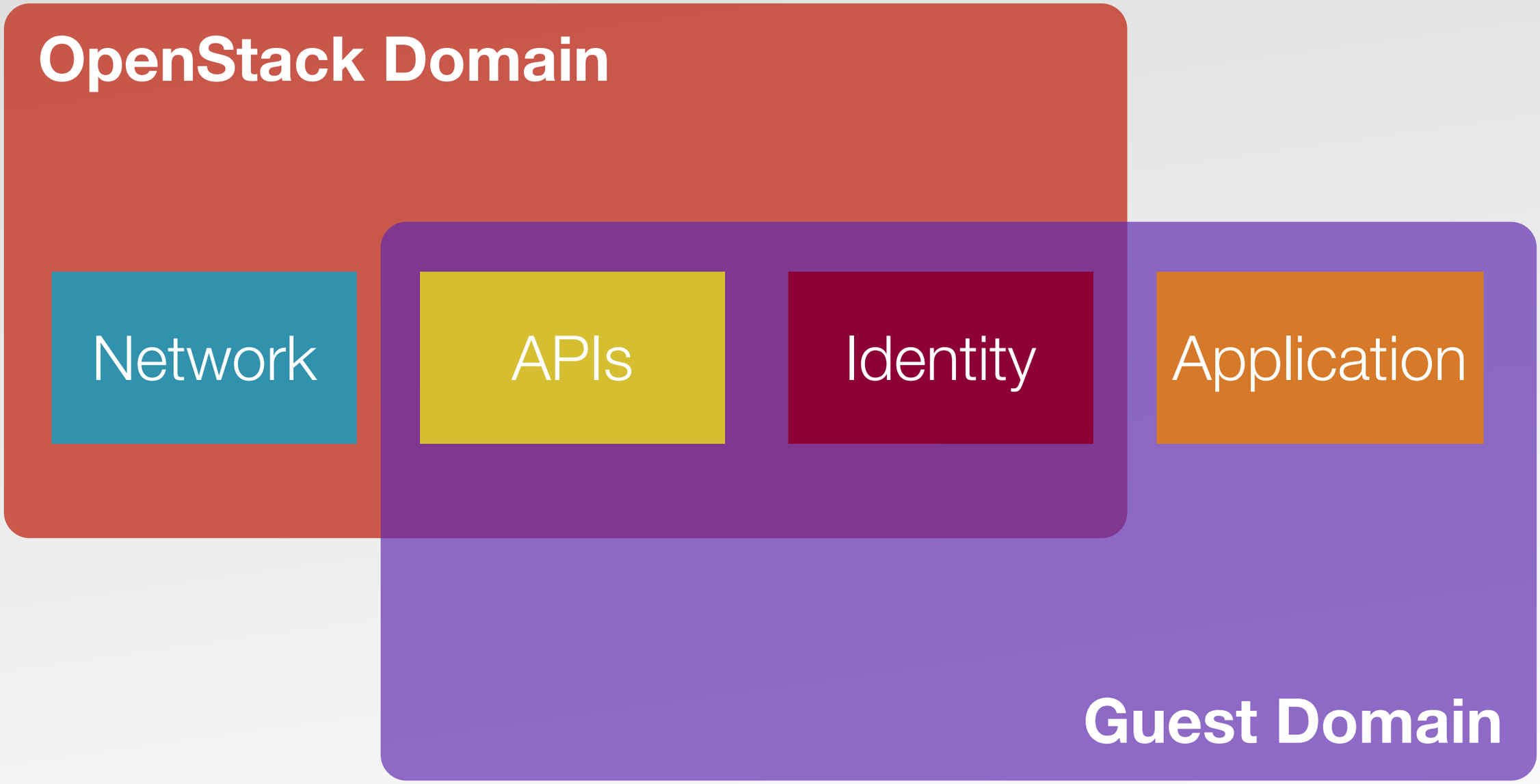
Exposed records as results of a data breach in the past 5 years[5]

8 months

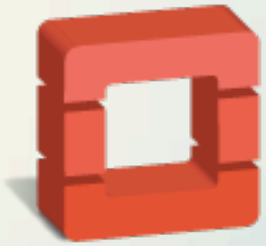
Is the average time an advanced threat goes unnoticed on victim's network[2]

1,3,5: Increased cyber security can save global economy trillions, McKinsey/World Economic Forum, January 2014 **2**: M-Trends 2013: attack the security gap, Mandiant, March 2013 **4**: ISACA's 2014 APT study, ISACA, April 2014. Source: ISACA Cyber Security Nexus

OpenStack and Guest Security



Network Security
(OpenStack built-in systems)



openstack™
CLOUD SOFTWARE

Linux Namespaces

Used in OpenStack, widely adopted in Neutron, it was Originally created for Linux Control Groups (aka cgroups)

PID namespaces

isolate the process ID number space so that processes in different PID namespaces can have the same PID

Mentioning:

IPC and Unix Time-Sharing (UTS) namespaces

User namespaces

isolate the user and group ID number spaces.

Mount namespaces

isolate the set of filesystem mount points seen by a group of processes.

Network namespaces

provide isolation of the system resources associated with networking

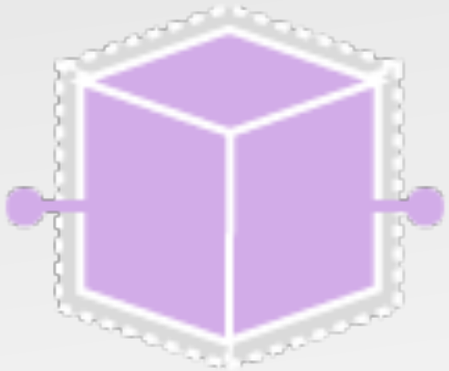
OpenStack Neutron

Software-Defined Network in OpenStack, it answer RESTful APIs.

Neutron Server runs on Controller, expose APIs, enforce network model, pass to Neutron Plugin

Neutron Plugin runs on Controller, implements APIs, every vendor can create its own “implementation” (ex: Cisco, Juniper, ...)

Plugin Agent, run on each compute node and connect instances to the virtual network



Default implementation based on **OpenVSwitch**

OpenFlow to be set as fundamental open protocol for building SDN

Still no “industry” standard for encapsulating VLANs over L3, VXLANs set to be a preferred choice but any vendor has its choice (ex: Juniper has MPLS over IP)

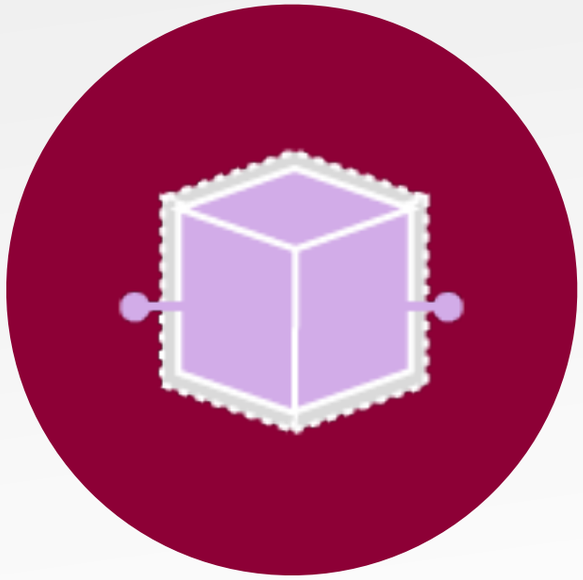
OpenStack Neutron and Network Namespaces

Namespaces enables **multiple instances of a routing table** to co-exist within the same Linux box

Network namespaces make it possible to separate network domains (network interfaces, routing tables, iptables) into completely separate and independent virtual datacenters

Advantage of namespaces implementation in Neutron is that tenants can create overlapping IP addresses and independent routing schema

The **neutron-l3-agent** is designed to use network namespaces to provide multiple independent virtual routers per node.



Example of Network Namespaces

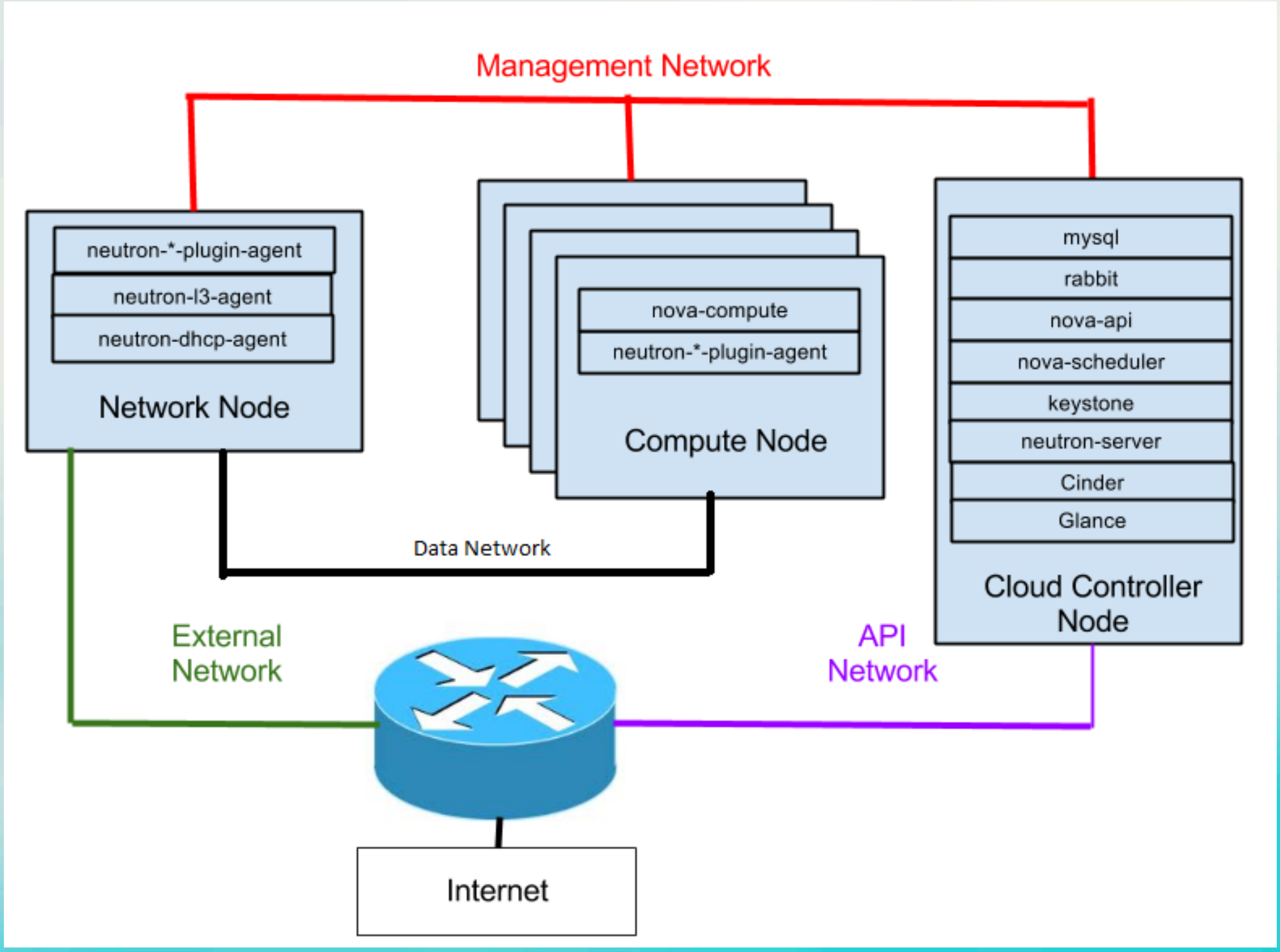
List Namespaces

```
# ip netns
qrouter-a88f89b6-5505-4bc2-8993-57ae1f010895
qdhcp-bebd6bc8-2bd0-4bdd-890c-9657faf80444
```

Show firewall rules in a virtual router

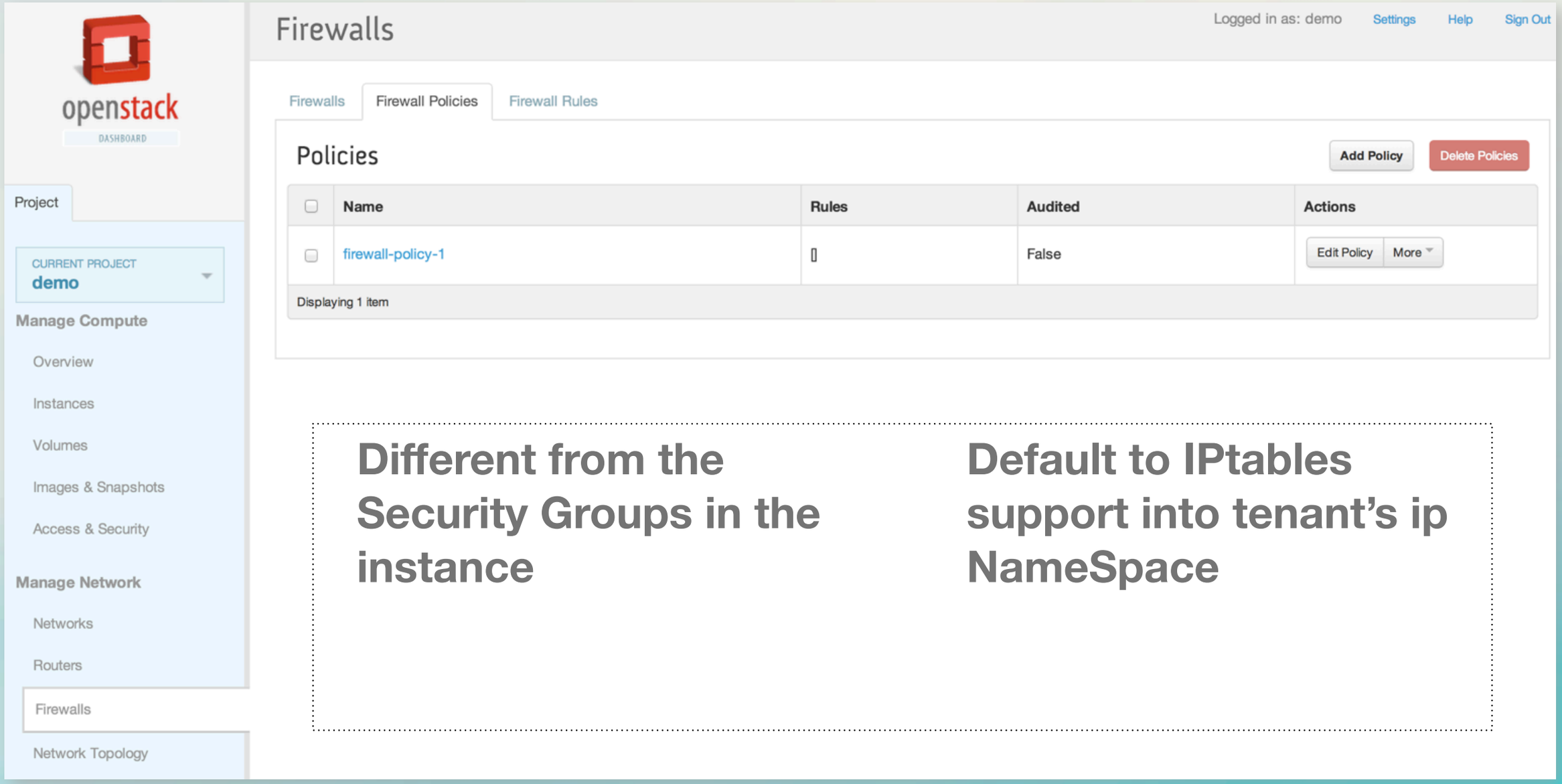
```
# ip netns exec qrouter-a88f89b6-5505-4bc2-8993-57ae1f010895
iptables -L -vn
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 neutron-l3-agent-INPUT  all  --  *      *
0.0.0.0/0           0.0.0.0/0
```

OpenStack Neutron L3 Agent



OpenStack Neutron FWaaS

Firewall as a Service in Neutron



Different from the Security Groups in the instance

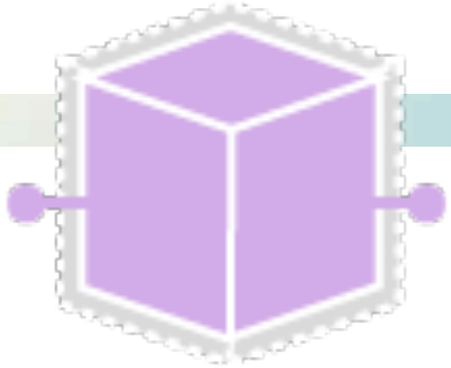
Default to IPtables support into tenant's ip NameSpace

OpenStack Neutron VPNaaS

Neutron has capability to handle per-tenant VPNs, named VPN-as-a-Service

Based on IPSec, just implementing IKE with “PSK” authentication mode rather than using certificates

Implemented on top of IP NameSpaces (“ip netns add vpn”)

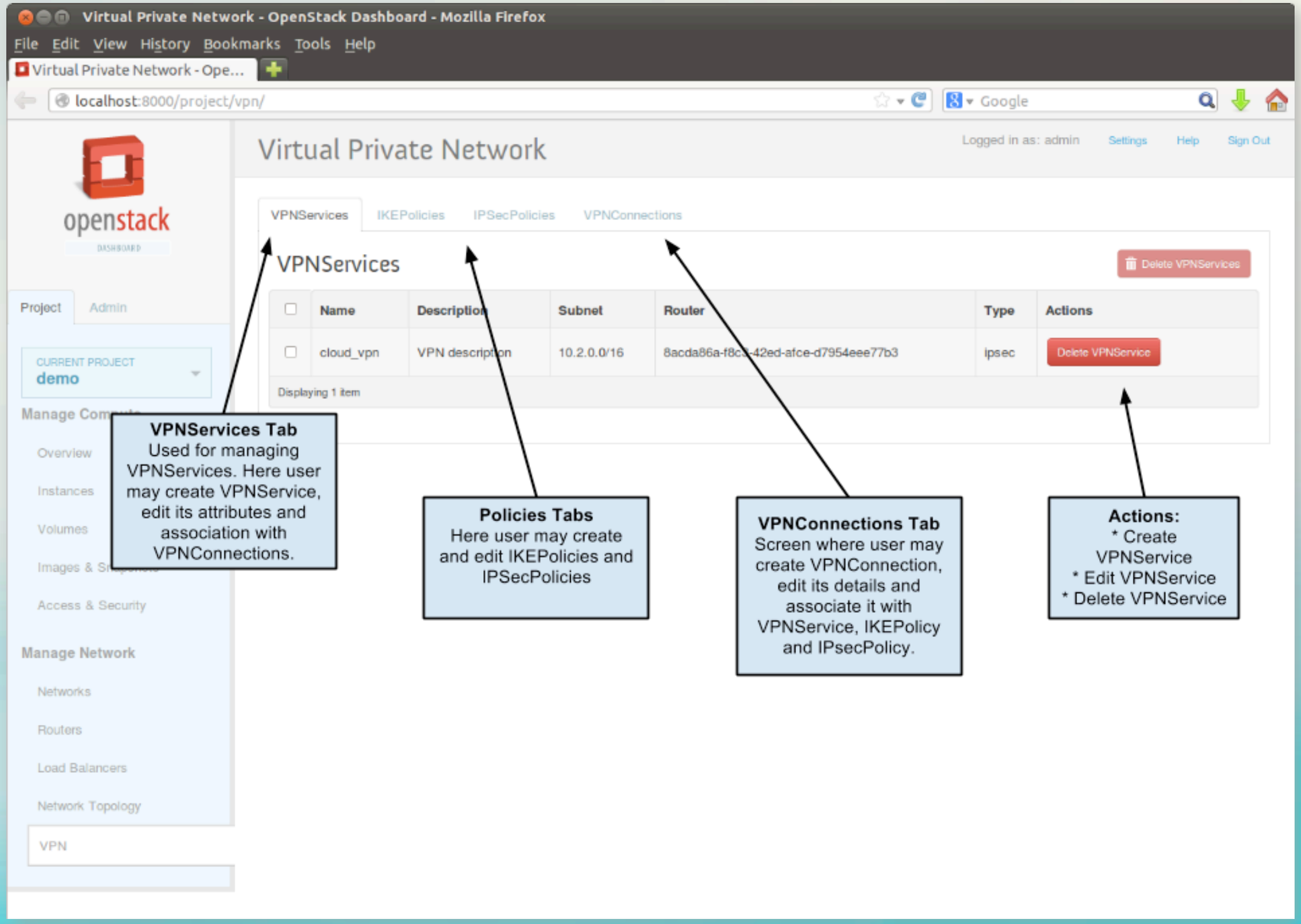


Draft exists on bringing OpenVPN to Neutron

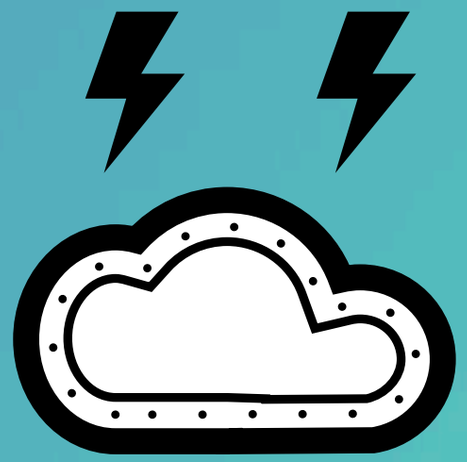
Suited for site-to-site VPNs and provide Hybrid cloud

Not suited for “roadwarriors”, i.e. clients connection

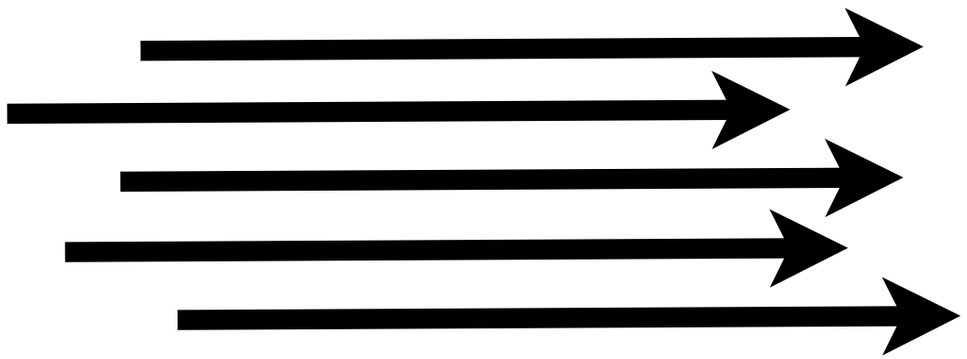
OpenStack Neutron VPNaaS



APIs Security
(OpenStack and Cloud Applications)



Web-based APIs



REST, XML-RPC, ...

APIs are your point of contact
from external world,
you must make them **highly secure**

Firewall are not enough!
Anything can be sent over HTTP/
HTTPS.

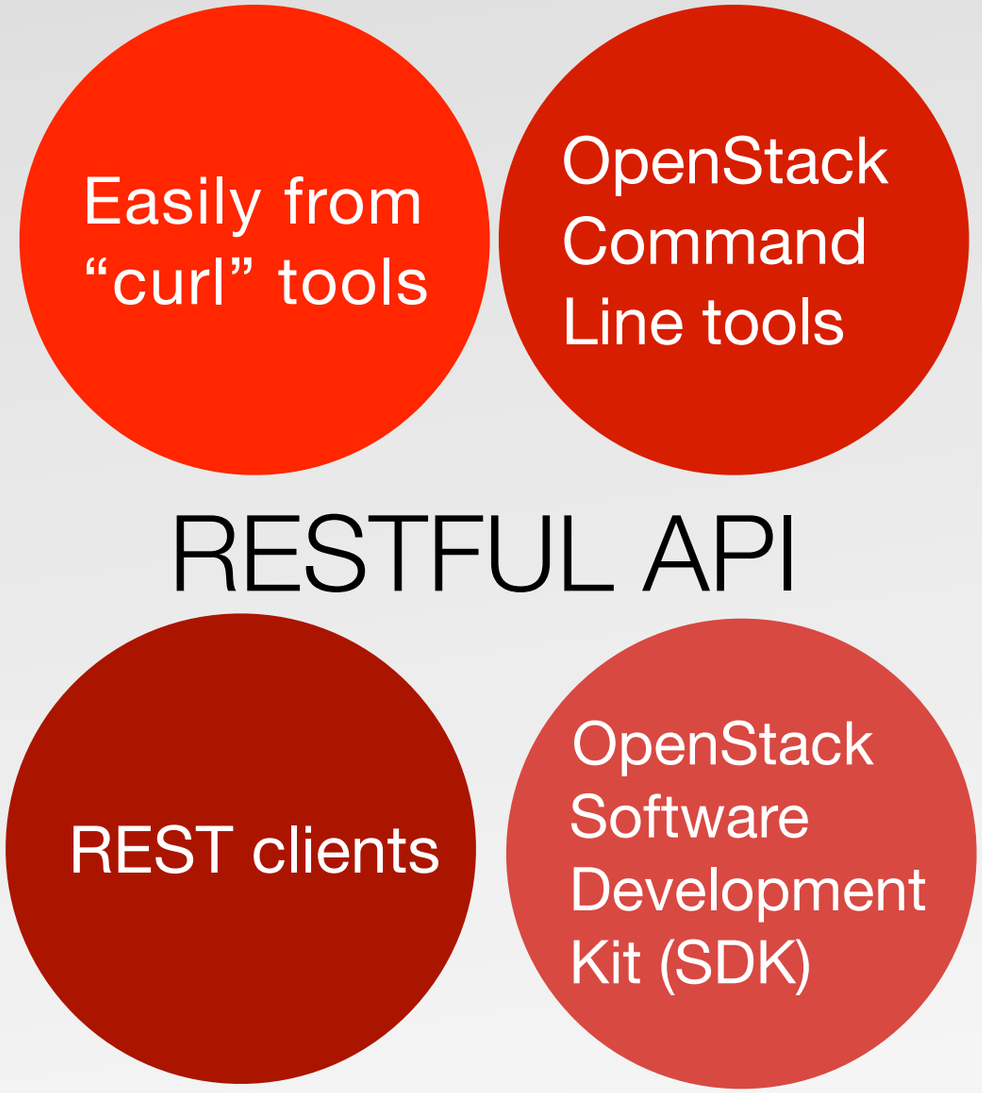
General APIs best practices

- Usernames and passwords, session tokens and API keys must **never** appear in the URL (Proxy caching and logging)
- Allow only selected HTTP methods
- Protect privileged actions and sensitive resource collections
- **Validate inputs** and enforce typing of values
- Validate incoming Content-Type and other headers
- **Encrypt data** in transit
- Validation also apply to payload: JSON, XML or whatsoever

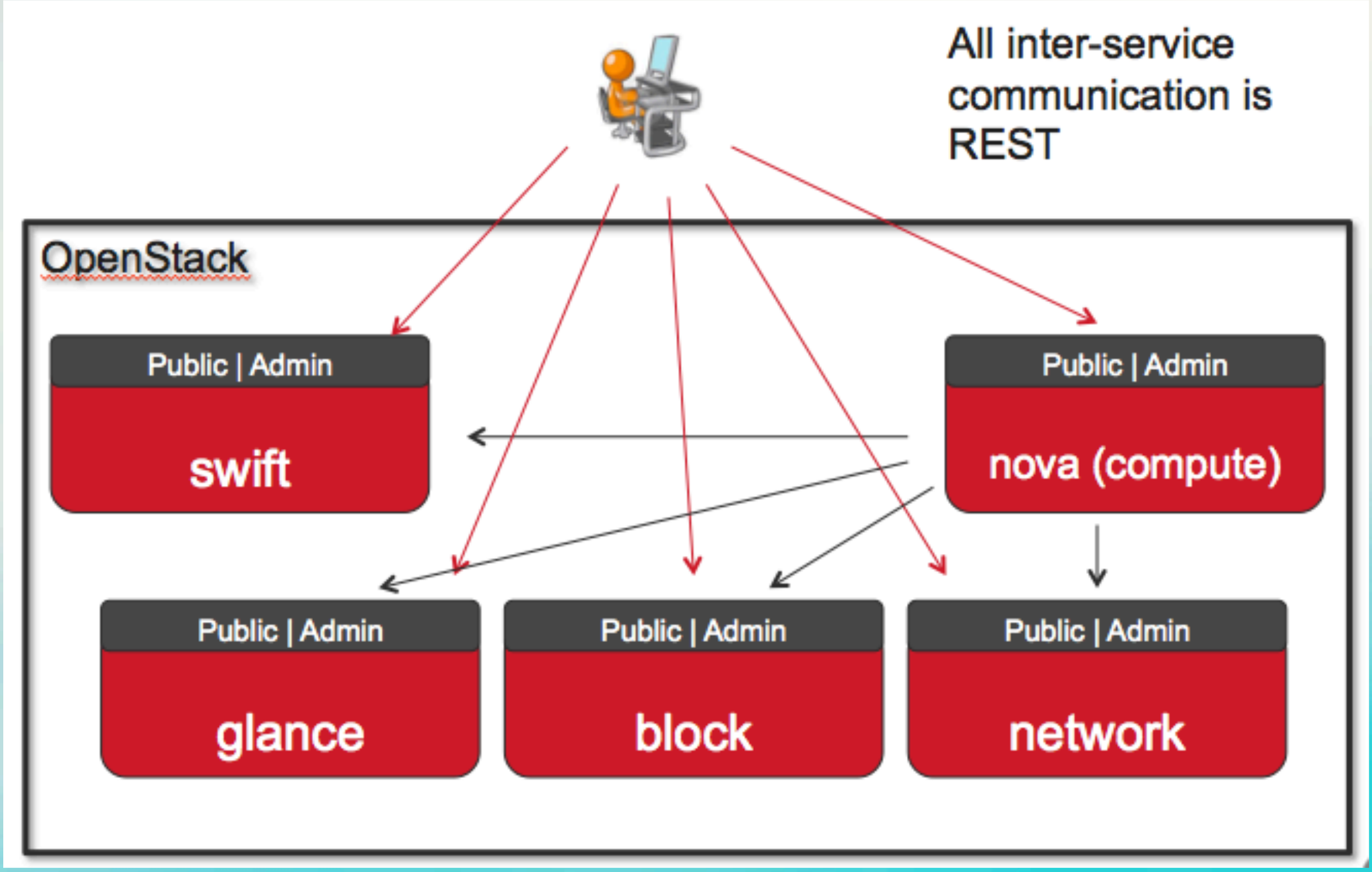
OpenStack APIs

All OpenStack software is based on APIs, consumed from End customers and tools to access the platform programmatically

Among OpenStack components, is a way of decoupling components implementations



OpenStack APIs EndPoints



OpenStack APIs Workflow

1. Obtain a Token

```
curl -d '{"auth":{"tenantName": "customer-x", "passwordCredentials":
{"username": "joeuser", "password": "secrete"}}}' -H "Content-type:
application/json" http://localhost:35357/v2.0/tokens
```

2. Consume the API (through the obtained token):

```
curl -i -X GET http://localhost:35357/v2.0/tenants -H "User-Agent:
python-keystoneclient" -H "X-Auth-Token: token"
```

Revealing the EndPoints

The token request will reveal the endpoints URLs: Compute/Nova, S3,Image/Glance, Volume/Cinder, EC2, Identity/Keystone


```
"serviceCatalog": [
    {
        "endpoints": [
            {
                "adminURL": "http://166.78.21.23:8774/v2/604bbe45ac7143a79e14f3158df67091",
                "region": "RegionOne",
                "internalURL": "http://166.78.21.23:8774/v2/604bbe45ac7143a79e14f3158df67091",
                "id": "9851cb538ce04283b770820acc24e898",
                "publicURL": "http://166.78.21.23:8774/v2/604bbe45ac7143a79e14f3158df67091"
            }
        ],
        "endpoints_links": [],
        "type": "compute",
        "name": "nova"
    },
    {
        "endpoints": [
            {
                "adminURL": "http://166.78.21.23:3333",
                "region": "RegionOne",
                "internalURL": "http://166.78.21.23:3333",
                "id": "0bee9a113d294dda86fc23ac22dce1e3",
                "publicURL": "http://166.78.21.23:3333"
            }
        ],
        "endpoints_links": [],
        "type": "s3",
        "name": "s3"
    },
```

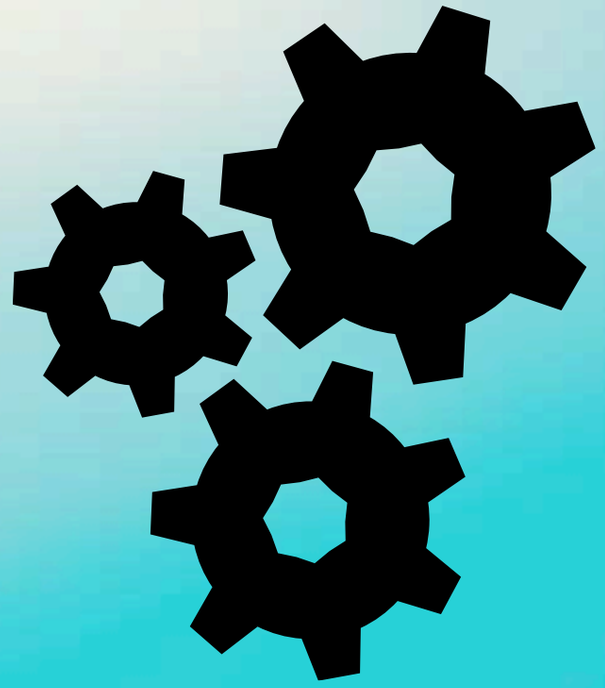
OpenStack APIs best practices

Isolate API endpoint processes, especially those that reside within the public security domain should be isolated as much as possible. API endpoints should be deployed on separate hosts for increased isolation.

Apply Defense-in-Depth concept: configure services, host-based firewalls, local policy (SELinux or AppArmor), and optionally global network policy.

Use Linux namespaces to assign processes into independent domains

Use network ACLs and IDS technologies to enforce explicit point to point communication between network services (ex: wire-level ACLs in L3 switches)



Mandatory Access Control in APIs

Isolate API endpoint processes from each other and other processes on a machine.

Use **Mandatory Access Controls (MAC)** on top of Discretionary Access Controls to segregate processes, ex: SE-Linux

Objective: **containment and escalation of API** endpoint security breaches.

Use of MACs at the OS level severely limit access to resources and provide earlier alerting on such events.

Ex: SecurePass NG (Dreamliner) APIs Security

RESTful APIs, mixture of POST (in request) and JSON (in response), Channel **encrypted** with TLS high cypher, Based on **APP ID** and **APP Secret**

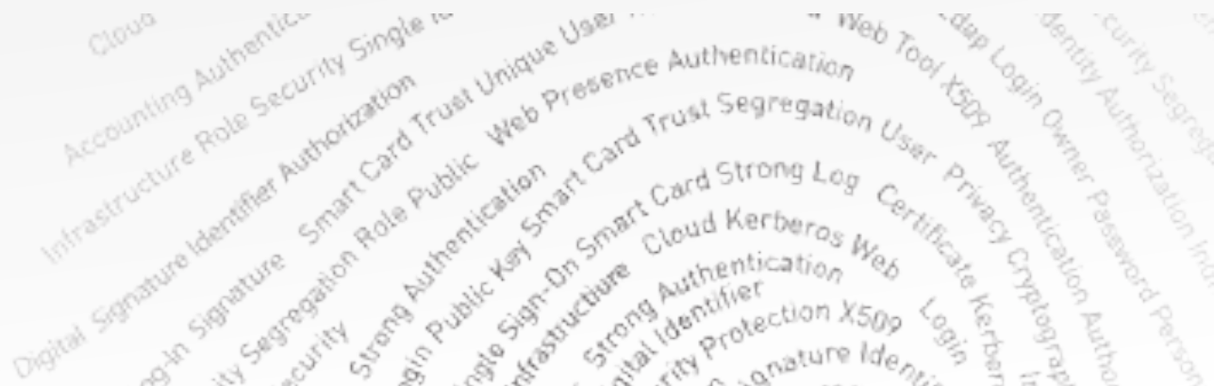
in *functionalities*, APP ID read-only or read-write

in *network*, APP ID can be limited to a given IPv4/IPv6

in *domain*, APP ID is linked to only a specific realm/ domain

Example: **/api/v1/users/info**

SECURE PASS™
Protecting Identities.

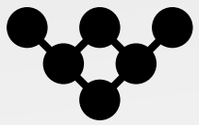


Identity Security
(OpenStack and Cloud Applications)

OpenStack Keystone

Provides Identity, Token, Catalog and policy services for uses inside the OpenStack family and implements **OpenStack's Identity APIs**

User management: keep tracks of users, roles and permissions



Service catalog: Provide a catalog of what services are available and where the OpenStack APIs EndPoint are located

OpenStack Identity Management

Users
A user represent a human user and has associated information such as username, password and e-mail

Tenants
A tenant can represent a customer, organization or a group.

Roles
A role is what operations a user is permitted to perform in a given tenant

Keystone permit the following back-ends for IDMs:
SQL Backend (SQLAlchemy, it's python), PAM, LDAP and custom plugins

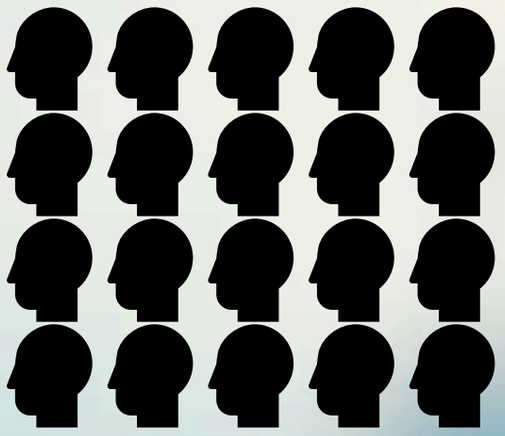
OpenStack Keystone

Catching username and passwords means reveal the whole OpenStack infrastructure and control it!

```
$ curl -d '{"auth":{"tenantName": "customer-x",
"passwordCredentials": {"username": "joeuser",
"password": "secrete"}}}' -H "Content-type: application/json"
http://localhost:35357/v2.0/tokens
```


```
"serviceCatalog": [
    {
        "endpoints": [
            {
                "adminURL": "http://166.78.21.23:8774/v2/604bbe45ac7143a79e14f3158df67091",
                "region": "RegionOne",
                "internalURL": "http://166.78.21.23:8774/v2/604bbe45ac7143a79e14f3158df67091",
                "id": "9851cb538ce04283b770820acc24e898",
                "publicURL": "http://166.78.21.23:8774/v2/604bbe45ac7143a79e14f3158df67091"
            }
        ],
        "endpoints_links": [],
        "type": "compute",
        "name": "nova"
    },
    {
        "endpoints": [
            {
                "adminURL": "http://166.78.21.23:3333",
                "region": "RegionOne",
                "internalURL": "http://166.78.21.23:3333",
                "id": "0bee9a113d294dda86fc23ac22dce1e3",
                "publicURL": "http://166.78.21.23:3333"
            }
        ],
        "endpoints_links": [],
        "type": "s3",
        "name": "s3"
    },
```

The victims of identity theft



10 millions
of victims of identity theft in USA in 2008
(Javelin Strategy and Research, 2009)

2 billions $
damages reported in Italy in 2009 (Ricerca ABI)

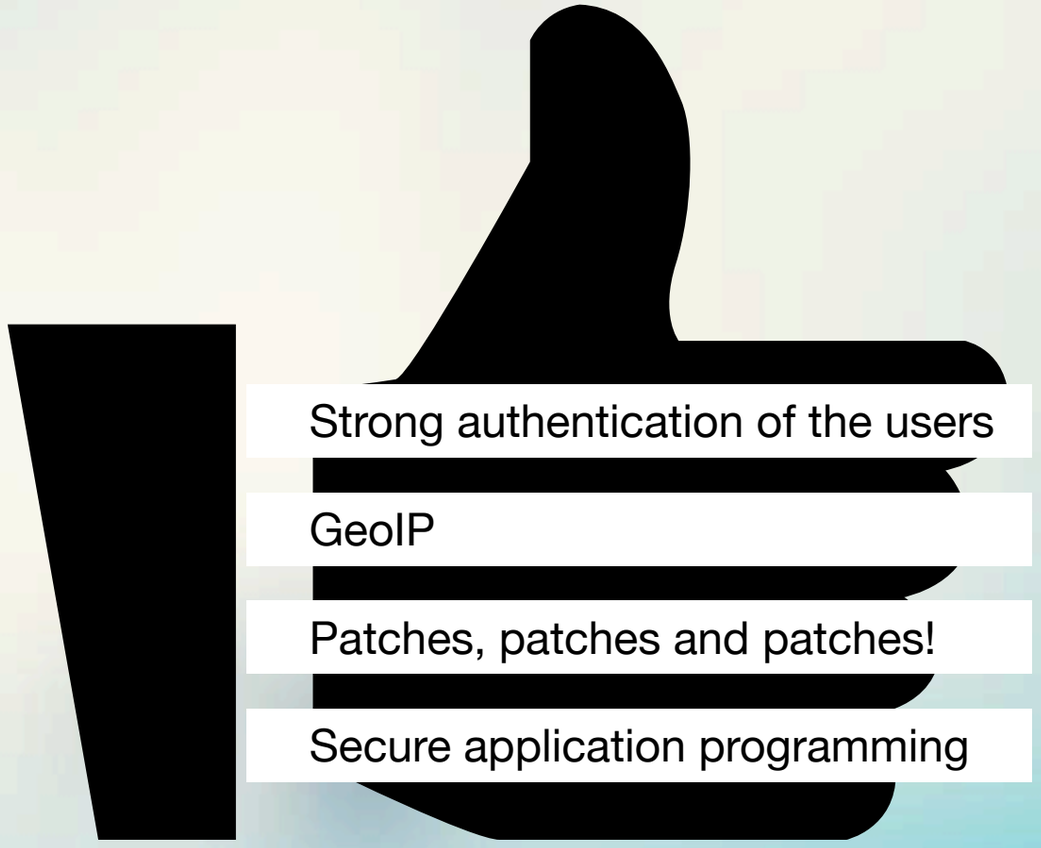
221 billions $
lost every year due to identity theft (Aberdeen Group)

35 billion
corporate and government records compromised in 2010
(Aberdeen Group)

2 years
of a working resource to correct damages due to identity theft (ITRC Aftermath Study, 2004)

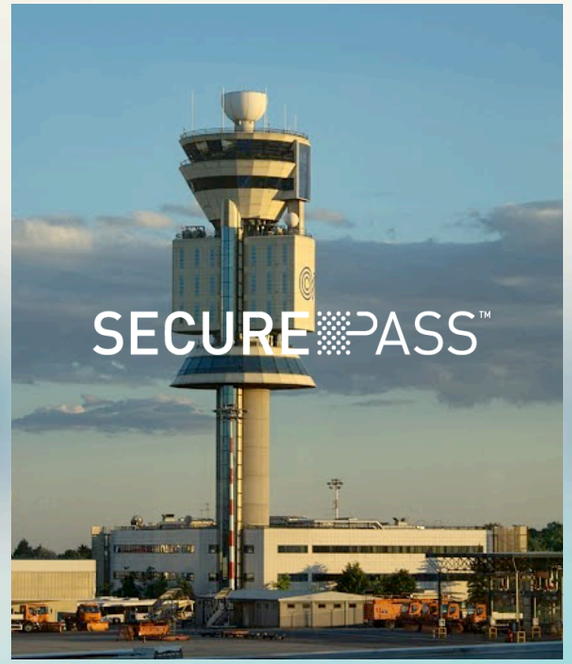
Identity best practices in applications

Security must be simple and transparent to the end user, otherwise it will be circumvented!



Need of a central Cloud Control

Cloud Orchestrator



2FA/SSO

Hosted Apps

Example of Web identity protection

```
<Directory /srv/www/myapp>

    AllowOverride None

    Order allow,deny

    allow from all

    AuthType CAS

    require spgroup mygroup@company.com

</Directory>
```

Require access through the SecurePass SSO portal with 2FA

Restrict to a dynamic group (with GeoIP)

Real-life example
(aka Case Study)

Case Study: Overview & Requirements

My accountant has his desktop computer broken, he has no time to change it, need something “always available” and in a restricted budget

He needs Windows for his accounting software

He has no office and works from home sometimes, he needs to access his desktop from ideally from his TV

He wants to connect from his customers’, but not always a computer available for him

He need emergency way of accessing the desktop from customers’ or from Internet Cafes (ex: on holidays)

Must provide a **secure access** as he holds very confidential data

Case Study: Solution

Virtualize his existing desktop system

From home, access the platform with an Android Mini-PC on existing HDMI TV, keyboard and a VPN with Mikrotik device
(Equipment ~120 EUR)

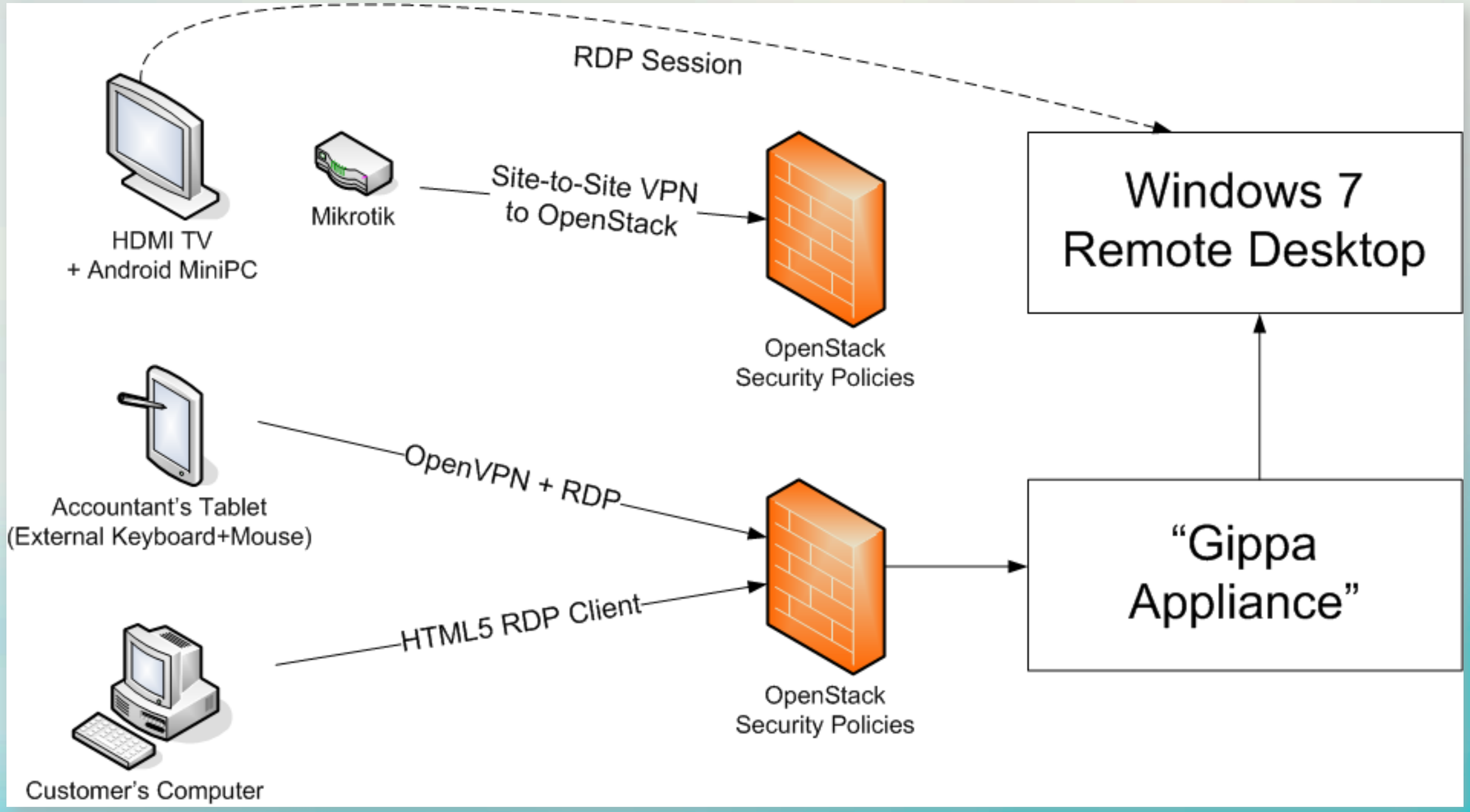
When at customer, access the platform with the existing Samsung Android tablet. Added bluetooth Keyboard + Mouse and OpenVPN
(K+M ~60 EUR)

Emergency access provided with an RDP HTML5 gateway

OpenStack as the operational platform

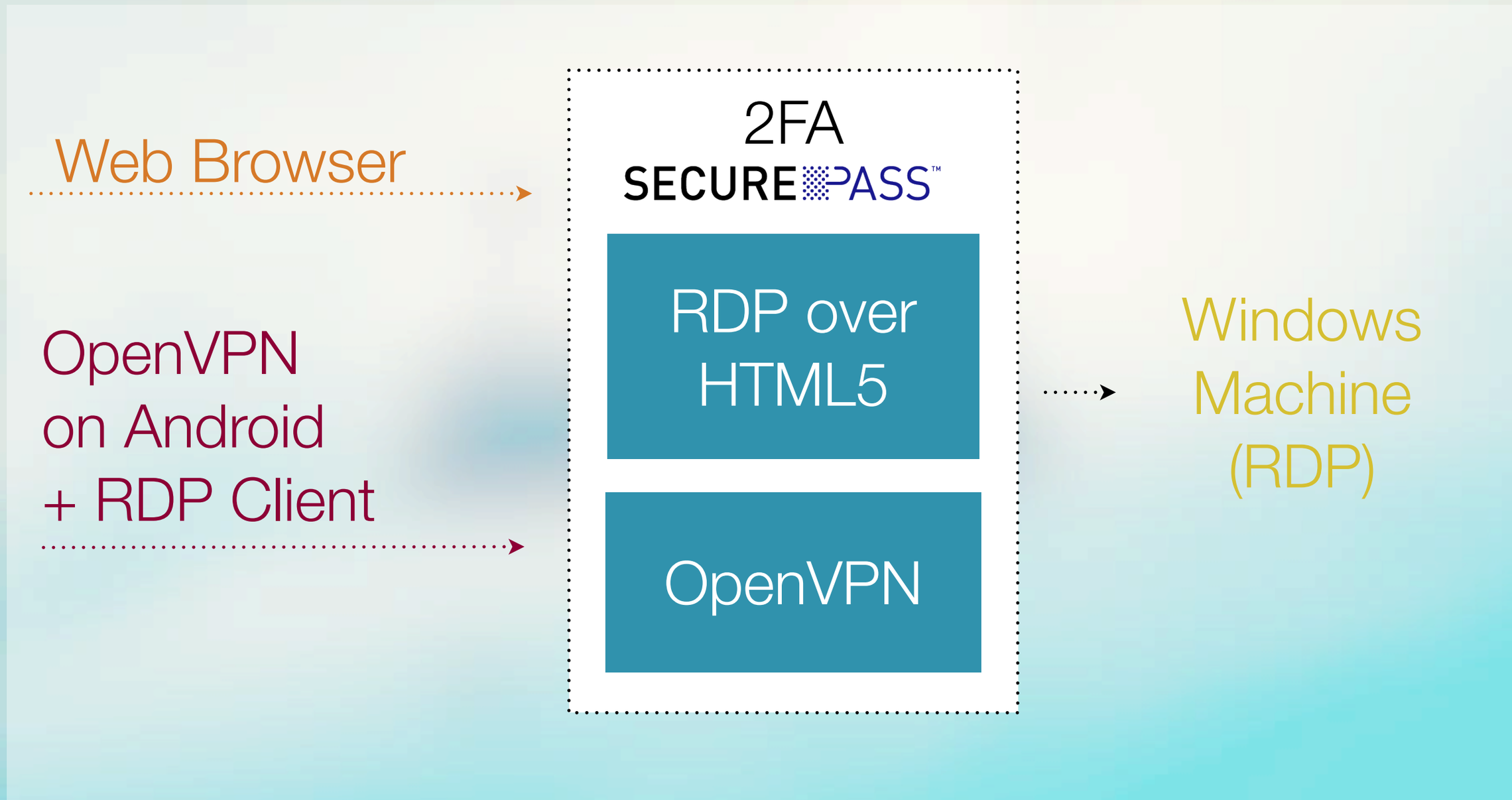
SecurePass as a security mechanism to protect access to his virtual desktop

Case Study: Overall Schema



Gippa

Appliance details



Acknowledgments

Security provided by

SECURE PASS™

www.secure-pass.net

Demo hosted by

OSTACK

powered by teuto.net
www.ostack.de

Thank you

GARL

MAKING THE CLOUD A SAFER SPACE