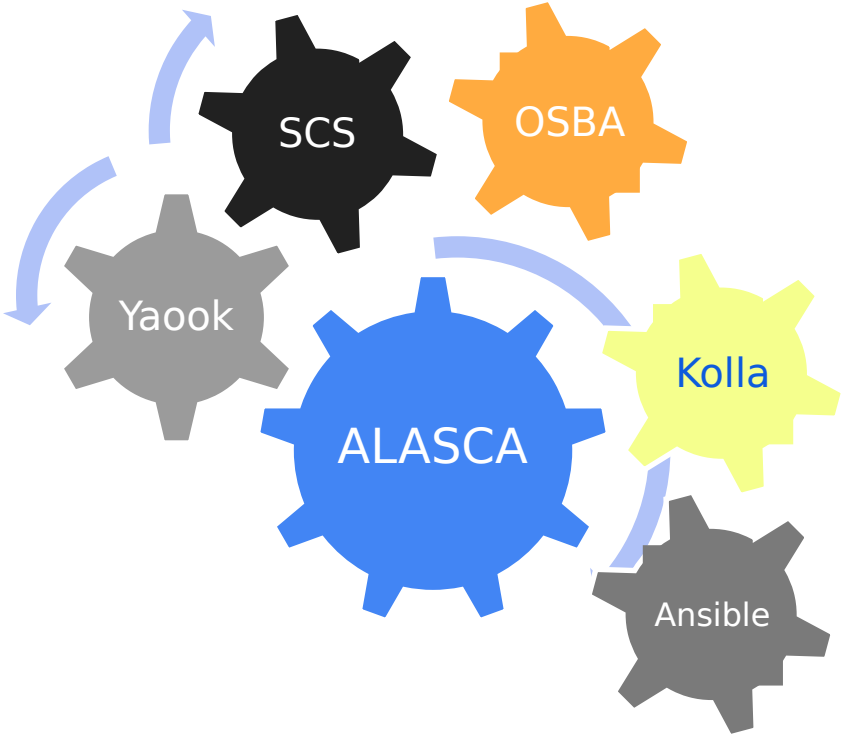
ALASCA

Verband für betriebsfähige, offene Cloud-Infrastrukturen e.V.

Sovereign Cloud Stack



Enhancing OpenStack Security Focus on Encryption

Martin Pilka, CEO

dNation MAKING CLOUD EASY

cloud@dNation.cloud | www.dNation.cloud

Developed as a part of SCS-VP04

- Tender: SCS-VP04 Networking
- Big Thanks to SCS, OSBA and BMWK

Challenge: OpenStack doesn‘t use Encryption Everywhere

- OpenStack uses encryption when talking to outside world
- However OpenStack internal components do not use encryption
- It is an issue for security sensitive application

Solution: E2E Encryption between OpenStack Internal Components

Two OpenStack LCM projects

1) Kolla Ansible

- Part of OpenStack
- Part of SCS efforts towards a standardized cloud stack

2) Yaook

- SCS certified implementation by ALASCA

Kolla Ansible Control Plane Security

- Goal: Allow operator to encrypt all channels using some simple settings, deploying certificates
- 18 SCS supported OpenStack services
- 6 additional system tools used: virtualization manager, database, cache, message broker, proxy, load balancer
- Identified all communication channels between them
- Analyzed state of encryption of these channel

Current State of Control Plane Encryption

Service/Peer	Backend	Caching	Database	Messaging	Intra-service	Inter-service
Keystone						
Horizon						
Glance						
Cinder						
Placement						
Nova						
Neutron						
Heat						
Octavia						
CloudKitty						
Barbican						
Designate						
Ceilometer						
Gnocchi						
Ironic						
Skyline						
Manila						
Masakari						
OpenvSwitch						
Redis						
libvirt						
Memcached						
MariaDB						
RabbitMQ						
	Existing support		New contributions		Not relevant	

SCS Contributions

- 20+ Kolla Ansible and Kolla projects contributions
- Simple settings
  - kolla_enable_tls_internal
  - kolla_enable_tls_external
  - kolla_enable_tls_backend

=> Whole control plane will use encrypted communication via TLS

Kolla Ansible Data Plane Security

Use case: Secure traffic between customer workloads when underlying infra can't be completely trusted, e.g. collocation

- Analyzed multiple approaches
- Summarized in ADR (Architecture Decision Record)
- End-to-End Encryption between Customer Workloads

End-to-End Encryption between Customer Workloads

Following options were analyzed

- TripleO + IPsec
- **OVN + IPsec: chosen option**
- Neutron + Cilium
- Neutron + Calico

OVN + IPsec: Transparent Encryption

openstack.

- IPsec with OVN logical networking
- Simple transparent solution, minimal setup necessary
- Dynamic - automatically reacts to changes in network
- Utilizes time tested IPsec solution
- Upstream contributions to Kolla and Kolla Ansible repositories

Yaook

Yaook Operator

Yaook Kubernetes

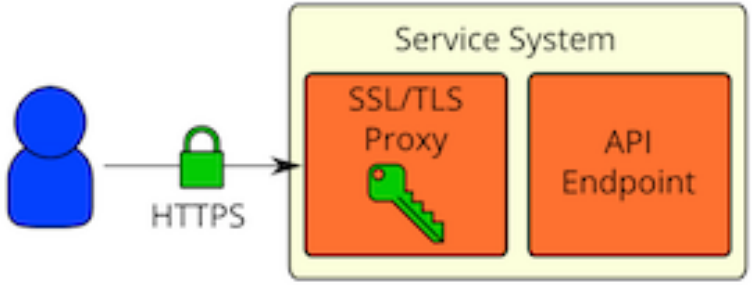
Yaook Bare metal

cloud@dNation.cloud | www.dNation.cloud

dNation MAKING CLOUD EASY

Yaook Encryption

Architecture: SSL/TLS on same physical hosts as API endpoints



Yaook Encryption

Each service has internal and public endpoints, e.g. *Keystone*:

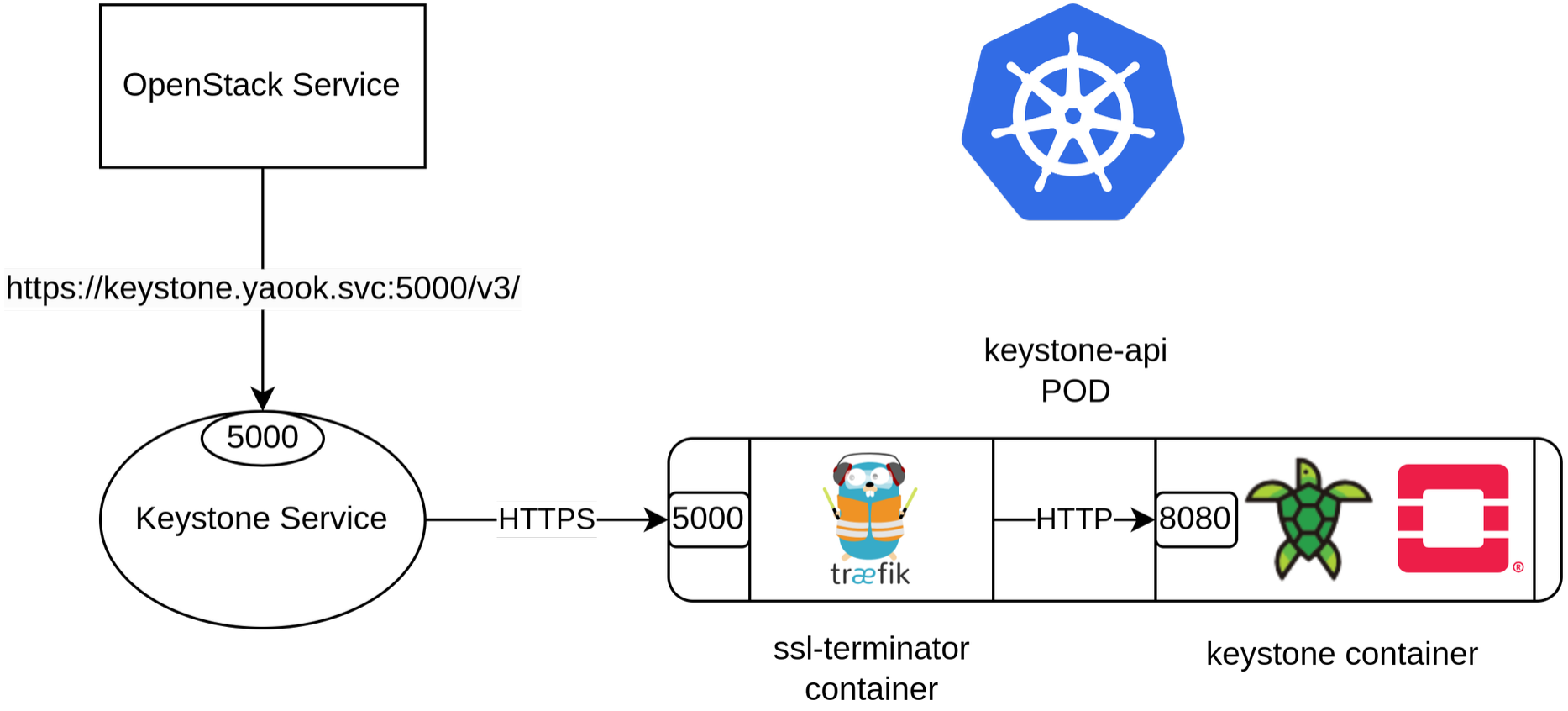
```
$ openstack catalog show keystone -c endpoints
+-----------+-------------------------------------------------------------+
| Field     | Value                                                       |
+-----------+-------------------------------------------------------------+
| endpoints | RegionOne                                                   |
|           |   internal: https://keystone.yaook.svc:5000/v3/             |
|           | RegionOne                                                   |
|           |   admin: https://keystone.yaook.svc:5000/v3/                |
|           | RegionOne                                                   |
|           |   public: https://keystone.regionone.dnation.cloud:443/v3/  |
|           |                                                             |
+-----------+-------------------------------------------------------------+
```

Yaook Encryption

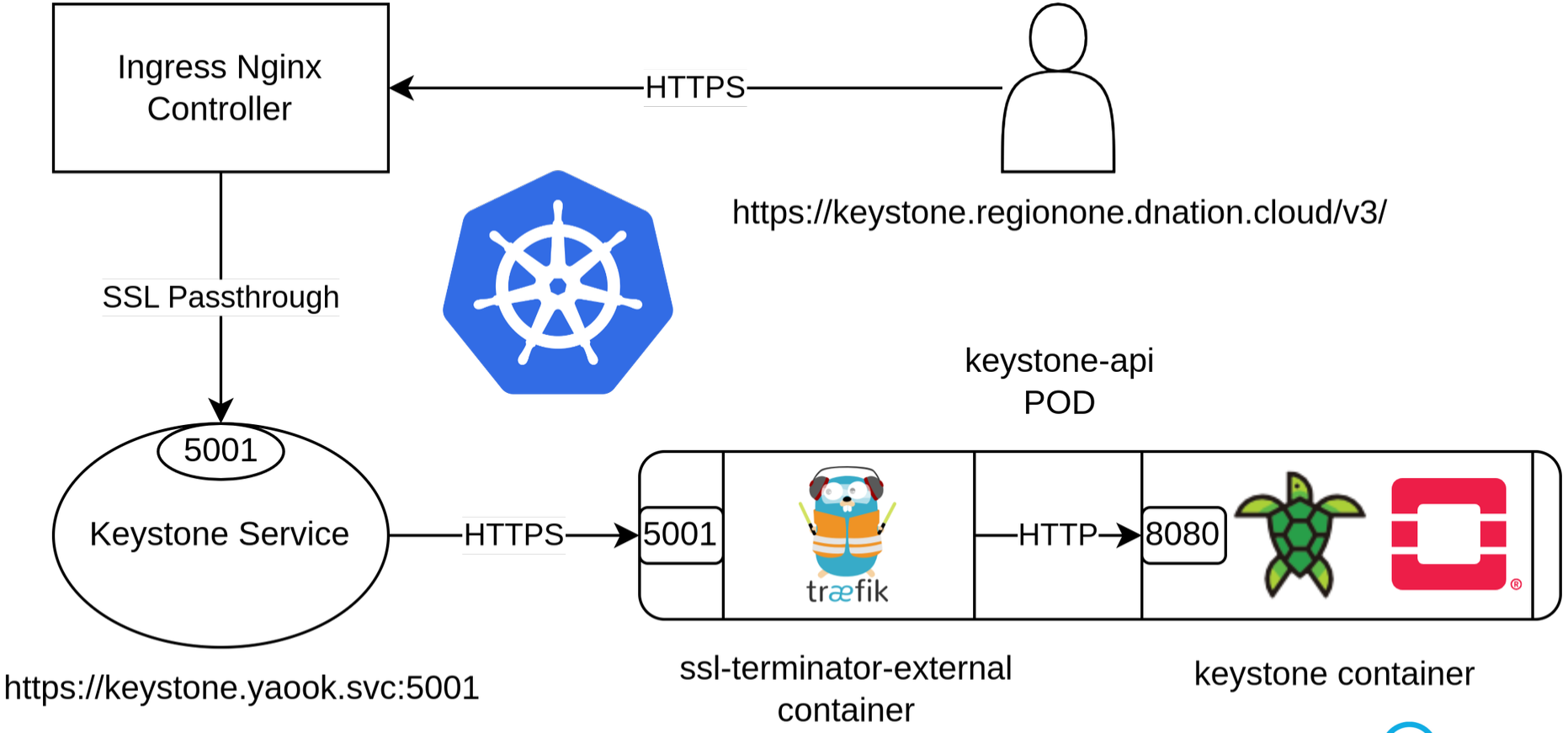
Each OpenStack service runs as Kubernetes POD with multiple containers:

- Primary Container - OpenStack service, e.g. *Keystone*
- SSL Terminator Container - Traefik Application Proxy terminates *internal* HTTPS traffic and passes it to the primary container
- External SSL Terminator Container - Traefik terminates *public* HTTPS traffic and passes it to the primary container

Internal Endpoint - Keystone Example



Public Endpoint - Keystone Example



Internal Traffic Encryption: OpenStack and Yaook

openstack.	
Kolla Ansible	Yaook Operator
OVN + IPsec	Kubernetes PODs with multiple containers
New	Battle tested in production

Additional work done as a part of SCS-VP04: SONiC

- **S**oftware for **O**pen **N**etworking **i**n the **C**loud

- Free and Open Source network operating system based on Linux
- Runs on Edge-core switches, e.g. AS7726-32X (100G switch)
- Decouples hardware and software, promoting interoperability in network solutions
- Supported by more than 100 switch platforms and all major ASICs
- Highly extensible through a container-based architecture and standardized APIs

- dNation is contributing to the upstream SONiC repositories through the SCS project to improve this open-source platform
- Utilize the community version of SONiC and set up a pure L3 (BGP) underlay by implementing the integrated FRR (Free Range Routing)
- Monitoring: Prometheus exporter for SONiC
  - Monsoon: Supports Broadcom Enterprise SONiC only
  - sonic-exporter: Community Prometheus exporter, only 1 contributor
  - Suits well dNation Monitoring

Mid-term goal: Dynamic SONiC configuration via Openstack

Thank you for your attention

cloud@dNation.cloud | www.dNation.cloud

dNation
MAKING CLOUD EASY