Fragility of API Interoperability

- Keep Open Source Interoperable -

Open Source Summit, Japan 2017

Ghanshyam Mann, NEC

Ghanshyam Mann
Software developer, NEC

- OpenStack upstream developer since 2014.

@ghanshyammann

https://www.linkedin.com/in/gmann1

https://github.com/ghanshyammann

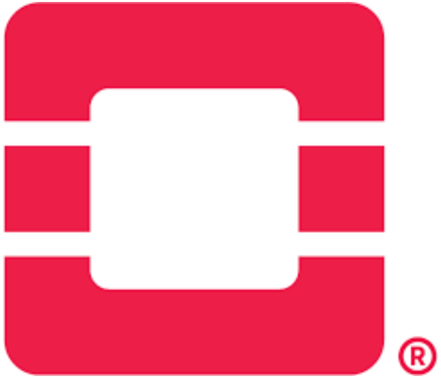
http://stackalytics.com/?release=all&user_id=ghanshyammann&metric=commits

REST API in Open Source

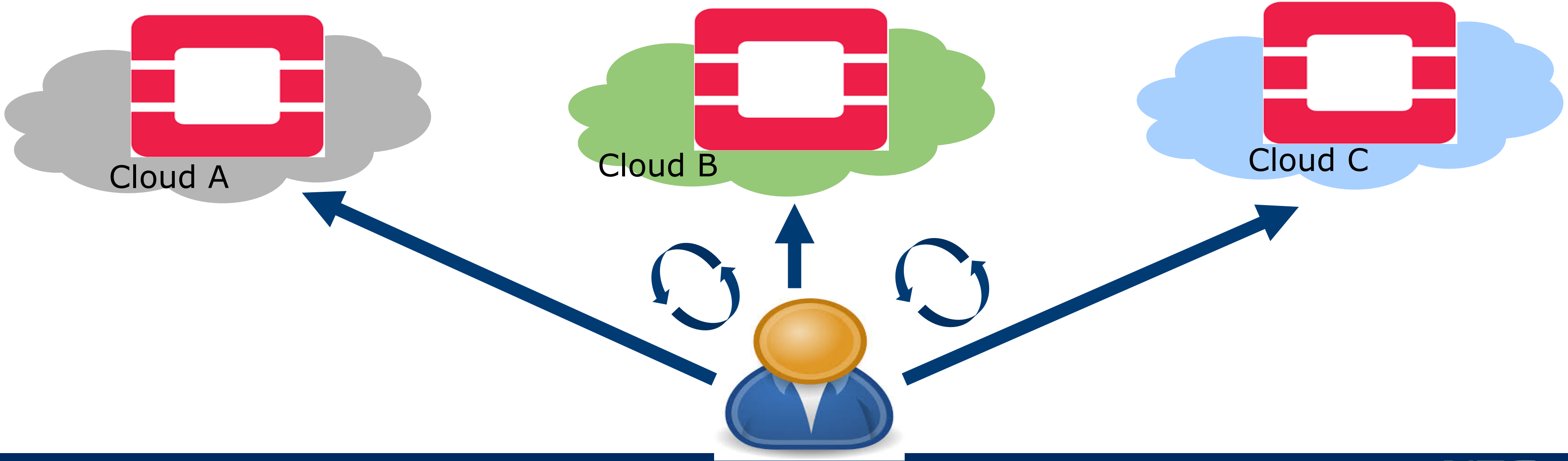
Today many OSS provide services through REST APIs

- Apache Mesos
- Kubernetes
- OpenStack

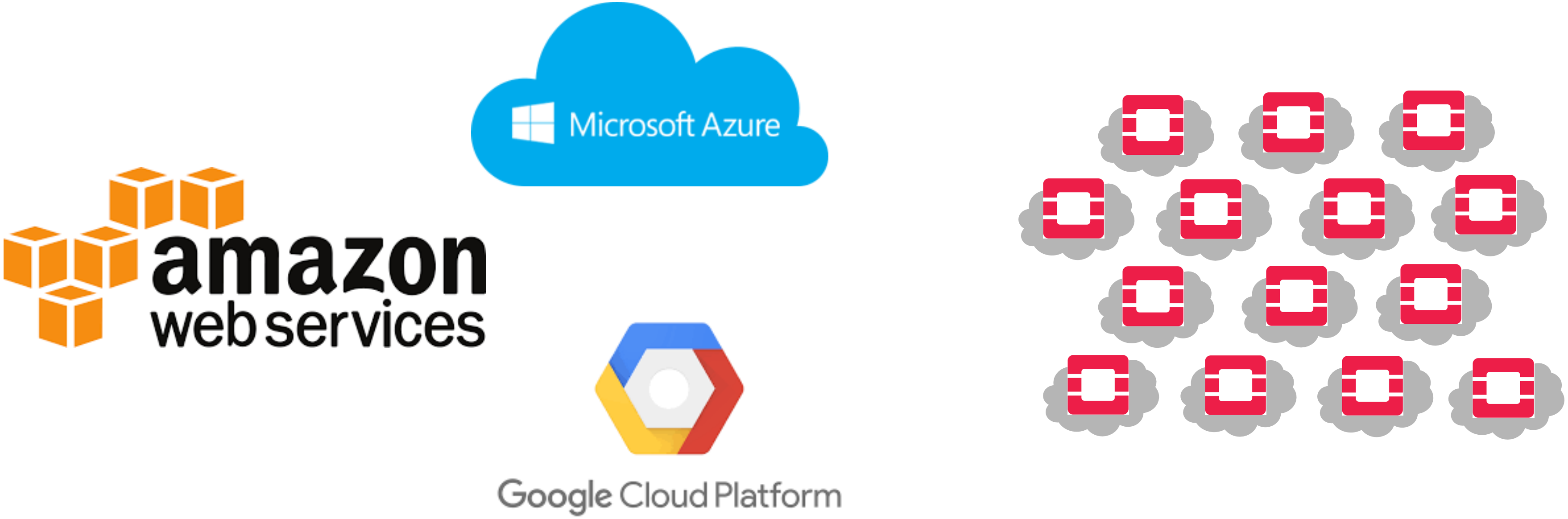
Users can extend their own APPs for catching business chances by using APIs



Interoperability



Why Interoperability is important?

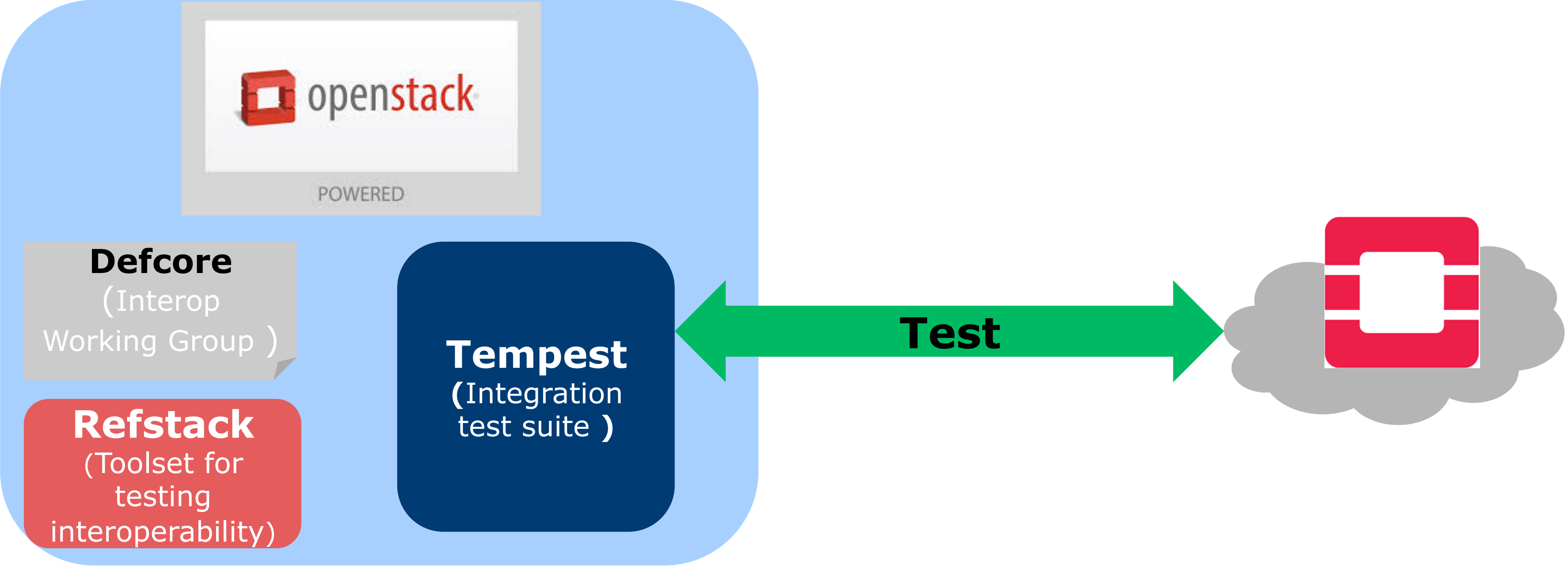


OpenStack interoperability

“OpenStack Powered”

- Certification program for OpenStack interoperability

How OpenStack Verify the Interoperability



Defcore
(Interop Working Group)

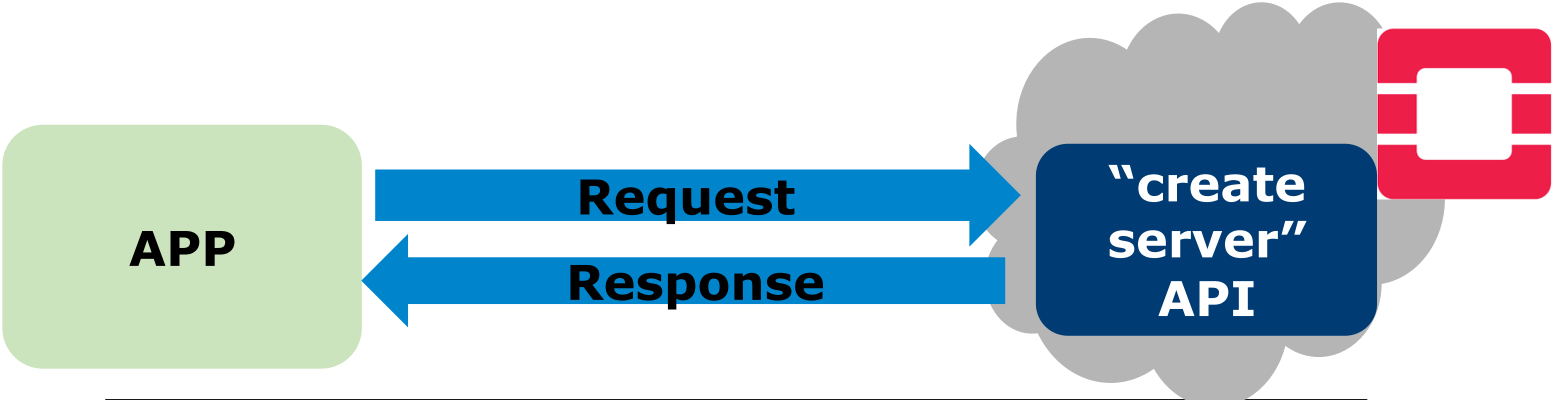
Refstack
(Toolset for testing interoperability)

Tempest
(Integration test suite **)**

Test

Backward Incompatibility & Interoperability

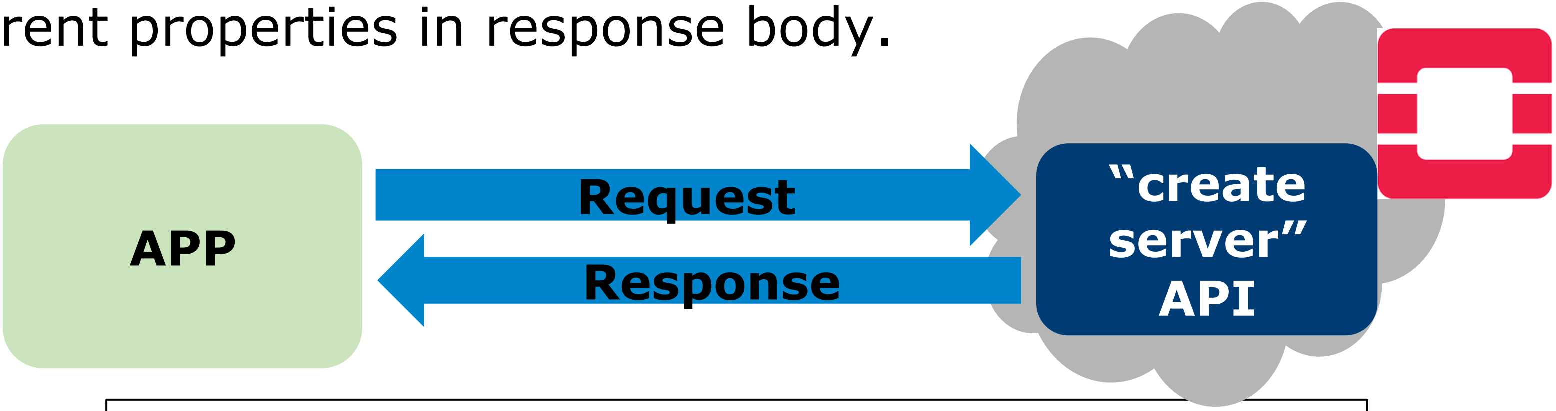
Example 1: Backward Incompatible



```
{"server": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "6NpUwoz2QDRN",
        "id": "f5dc173b-6804-445a-a6d8-c705dad5b5eb",
        "links_CHANGED_NAME": [..],
        "security_groups": []
}}
```

Example 2: Backward Compatible ??

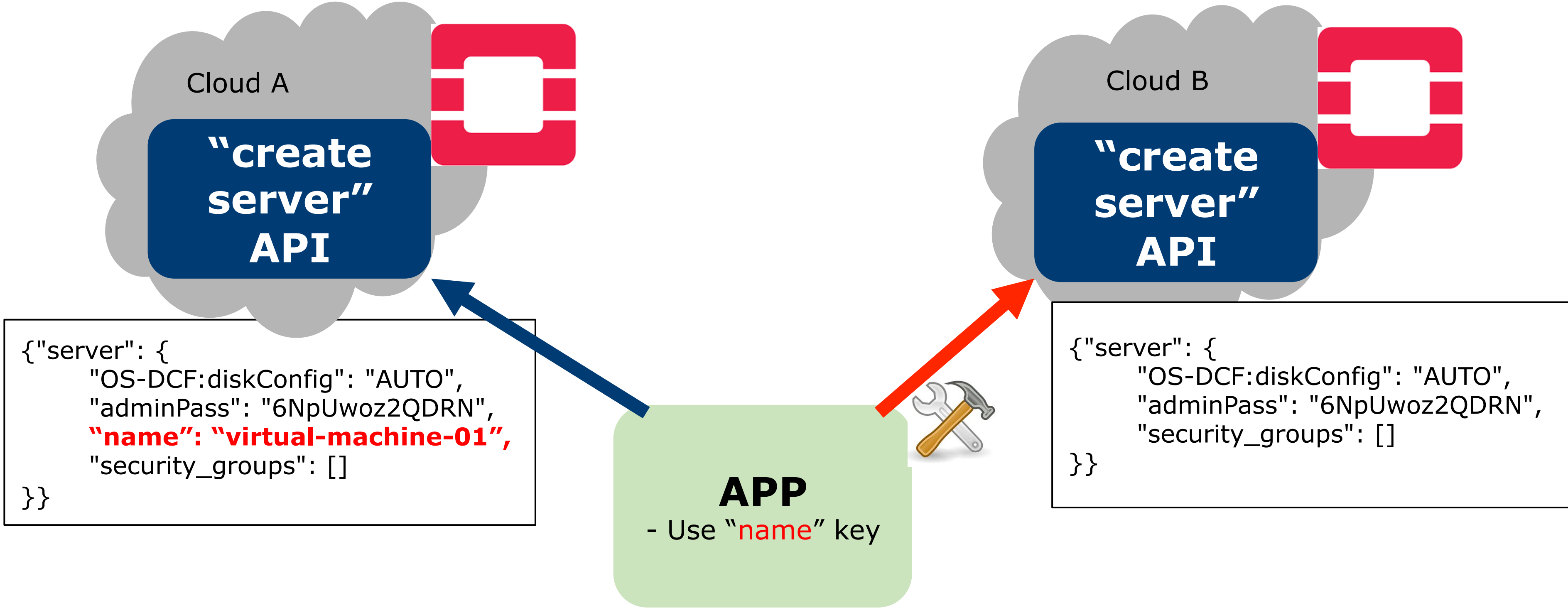
- Some cloud providers customize OpenStack API by adding different properties in response body.



```
{"server": {
       "OS-DCF:diskConfig": "AUTO",
       "adminPass": "6NpUwoz2QDRN",
       "id": "f5dc173b-6804-445a-a6d8-c705dad5b5eb",
       "links": [..],
       “name”: “virtual-machine-01”,
       "security_groups": []
}}
```

Example 2: Break Interoperability

- Some cloud providers customize OpenStack API by adding different properties in response body.



```
{"server": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "6NpUwoz2QDRN",
        “name”: “virtual-machine-01”,
        "security_groups": []
}}
```

```
{"server": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "6NpUwoz2QDRN",
        "security_groups": []
}}
```

APP

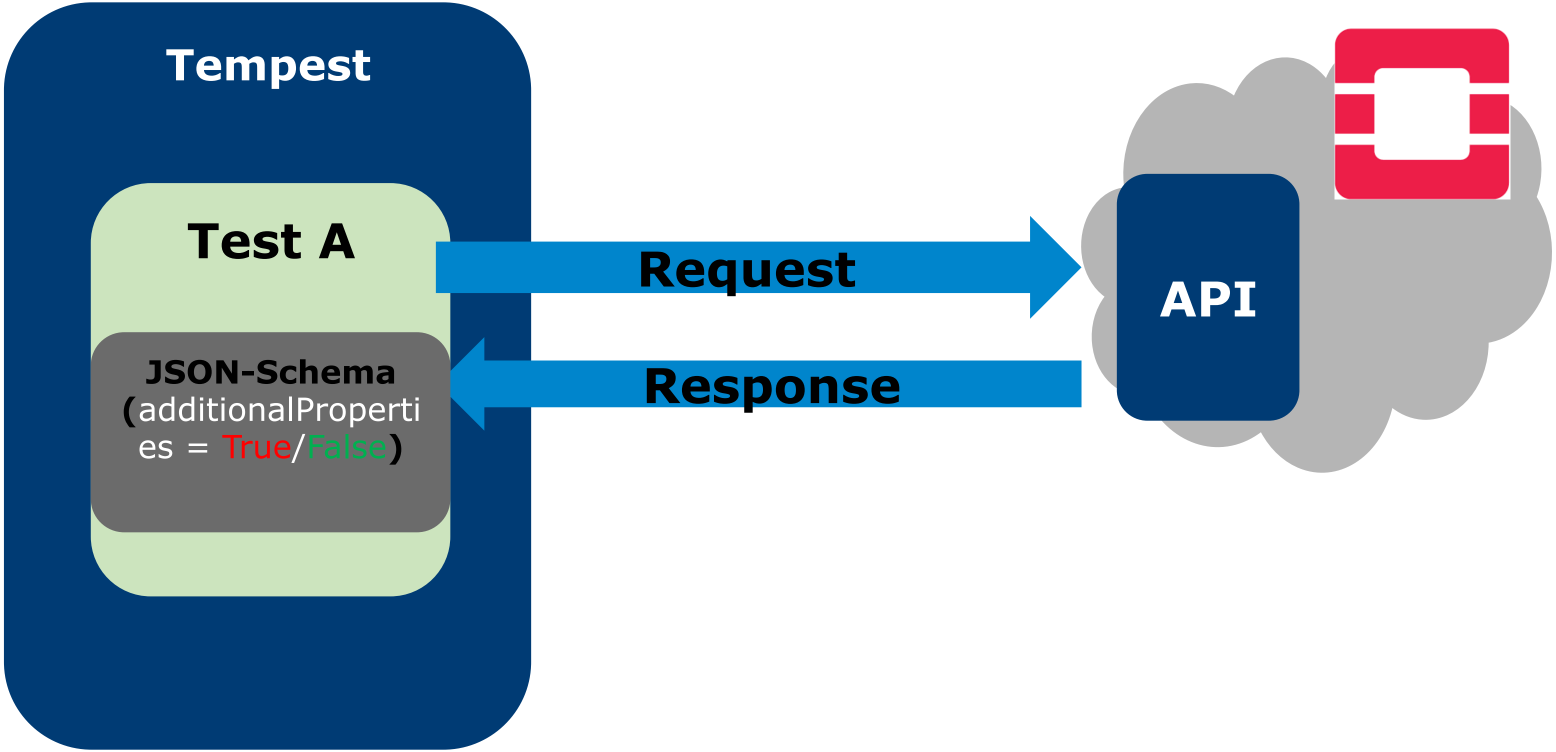
- Use “name” key

Example 2: Break Interoperability…

- https://review.openstack.org/#/c/467999/1

- It can be fixed on v2.1 API with microversion but fixing on v2 API is questionable ☹

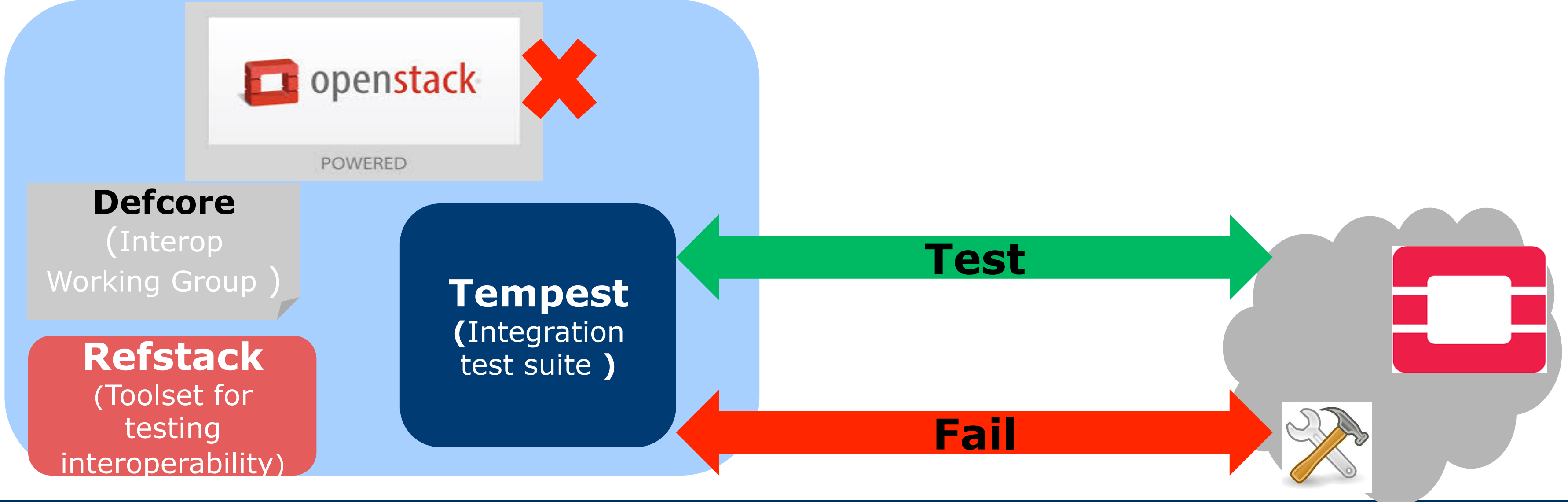
How Tempest Detect additional elements



Upstream First

Interoperability is not one Time things.

- Cloud got certification from OpenStack.
- But after few months/release, clouds failed to pass “OpenStack Powered”:
  - Because Tempest started failing when detecting customized APIs.
  - OpenStack APIs as well as Test is modified.



Defcore
(Interop Working Group)

Refstack
(Toolset for testing interoperability)

Tempest
(Integration test suite)

Test

Fail

Tempest JSON Schema

```
create_server = {
    'status_code': [202],
    'response_body': {
        'type': 'object',
        'properties': {
            'server': {
                'type': 'object',
                'properties': {
                    'id': {'type': 'string'},
                    .
                    .
                },
                'required': ['id']
            }
        },
        'required': ['server']
    }
}
```

```
create_server = {
    'status_code': [202],
    'response_body': {
        'type': 'object',
        'properties': {
            'server': {
                'type': 'object',
                'properties': {
                    'id': {'type': 'string'},
                },
                'additionalProperties': False,
                'required': ['id']
            }
        },
        'additionalProperties': False,
        'required': ['server']
    }
}
```

Discuss, discuss and discuss

- *Why we cannot pass this test, we should pass because we have never change anything since we passed it before.*

- *Such changes should be allowed on the certification test, most users use OpenStack SDKs to implement their own APPs and these SDKs just ignore these properties.*

- *If so, cloud providers don’t need to extend their own properties to API because user APPs just ignore.*

As the result

- We have implemented tool for distinguishing tests failed due to such extra properties changes or not.
- The certification program shows which APIs are customized based on this tool result.
- We have gray area of certification now, but we have a consensus to make all APIs the same for the interoperability.

Interoperability is Fragile

- As a developer, backwards compatible changes are not painful and we need to do it for improving/extending features.

- But it might break interoperability between cloud providers which are implemented based on Open Source.

- This principle made a lot of confusions in OpenStack community, but we have gotten a consensus about it today.

- By keeping interoperability, we can get more contributions from many people and make OSS better, because we need to implement additional features at upstream development.

How to keep Interoperability

JSON-Home

- JSON-Home provides available API resource URLs and HTTP methods(GET, PUT, etc) on each URL.

- Users get JSON-Home data by specifying

application/json-home

- We could expect users can implement their own APPs based on the JSON-Home data.

JSON-Home

GET /v2.1

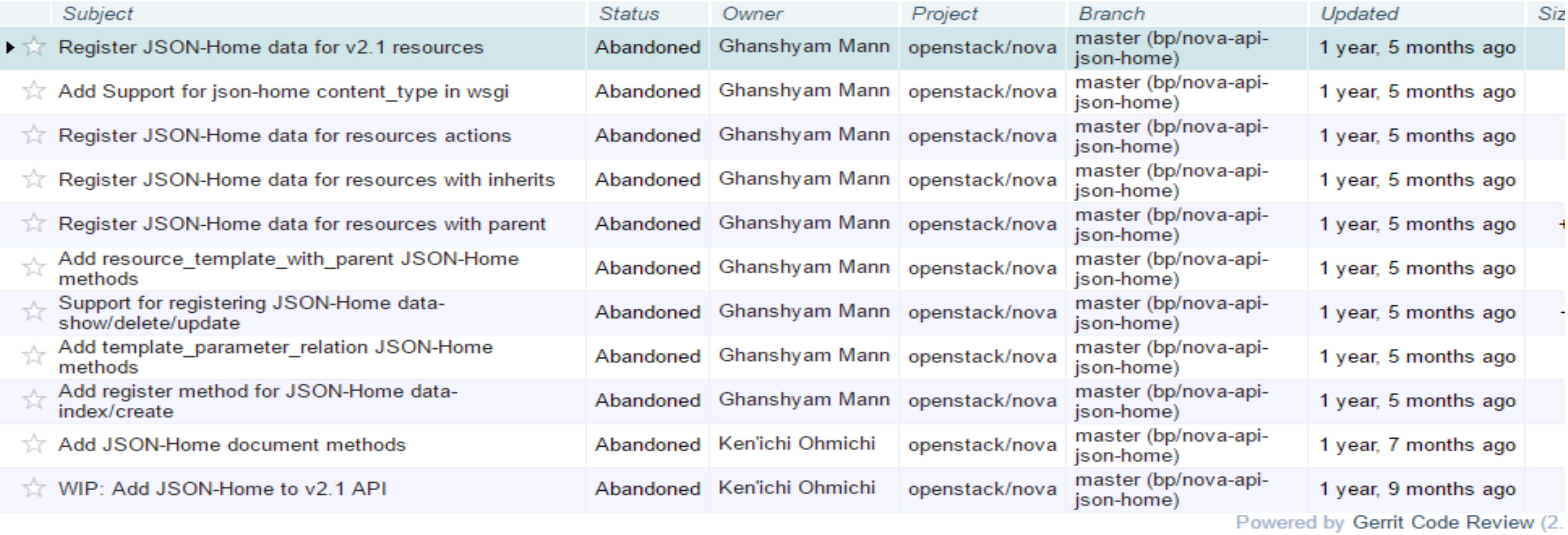
```
{
 "resources": {
   "http://docs.openstack.org/api/openstack-compute/2.1/rel/servers": {
     "href-template": "/v2.1/{project_id}/servers",
     "href-vars": {
       "project_id": "http://docs.openstack.org/api/openstack-compute/2.1/param/project_id"
     }
   },
   [..]
 }
}
```

GET /v2.1/{project_id}/servers

```
{
 "resources": {
   "http://docs.openstack.org/api/openstack-compute/2.1/rel/servers": {
     "href-template": "/v2.1/{project_id}/servers",
     "href-vars": {
       "project_id": "http://docs.openstack.org/api/openstack-compute/2.1/param/project_id"
     },
     "hints": {
       "allow": ["GET", "POST", "PUT", "DELETE"],
       "formats": {
           "application/json": {}
       },
       "accept-post": ["application/json"],
       "accept-put": ["application/json"]
     }
   }
 }
}
```

JSON-Home

- However, JSON-Home doesn't cover request/response body schema and we could not use it if not enhancing JSON-Home spec.



Subject	Status	Owner	Project	Branch	Updated	Siz
Register JSON-Home data for v2.1 resources	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	
Add Support for json-home content_type in wsgi	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	
Register JSON-Home data for resources actions	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	
Register JSON-Home data for resources with inherits	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	
Register JSON-Home data for resources with parent	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	+
Add resource_template_with_parent JSON-Home methods	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	
Support for registering JSON-Home data-show/delete/update	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	-
Add template_parameter_relation JSON-Home methods	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	
Add register method for JSON-Home data-index/create	Abandoned	Ghanshyam Mann	openstack/nova	master (bp/nova-api-json-home)	1 year, 5 months ago	
Add JSON-Home document methods	Abandoned	Ken'ichi Ohmichi	openstack/nova	master (bp/nova-api-json-home)	1 year, 7 months ago	
WIP: Add JSON-Home to v2.1 API	Abandoned	Ken'ichi Ohmichi	openstack/nova	master (bp/nova-api-json-home)	1 year, 9 months ago	

Powered by Gerrit Code Review (2.

OpenAPI (Swagger)

- OpenAPI covers request/response body schema.
- OpenAPI provides API spec with the detail through REST API to users.
  - We can expect users can implement their own APPs from the API spec automatically.
  - OpenStack community also defined OpenAPI as one of standard ways, but no components implemented it yet.
- OpenAPI is used by Kubernetes, Google, IBM Watson and lot more.

OpenStack API Microversion

- Microversion is OpenStack specific mechanism.
- Every small API change is versioned.
- Microversion are per API request.
- Users can choose necessary version on API header.

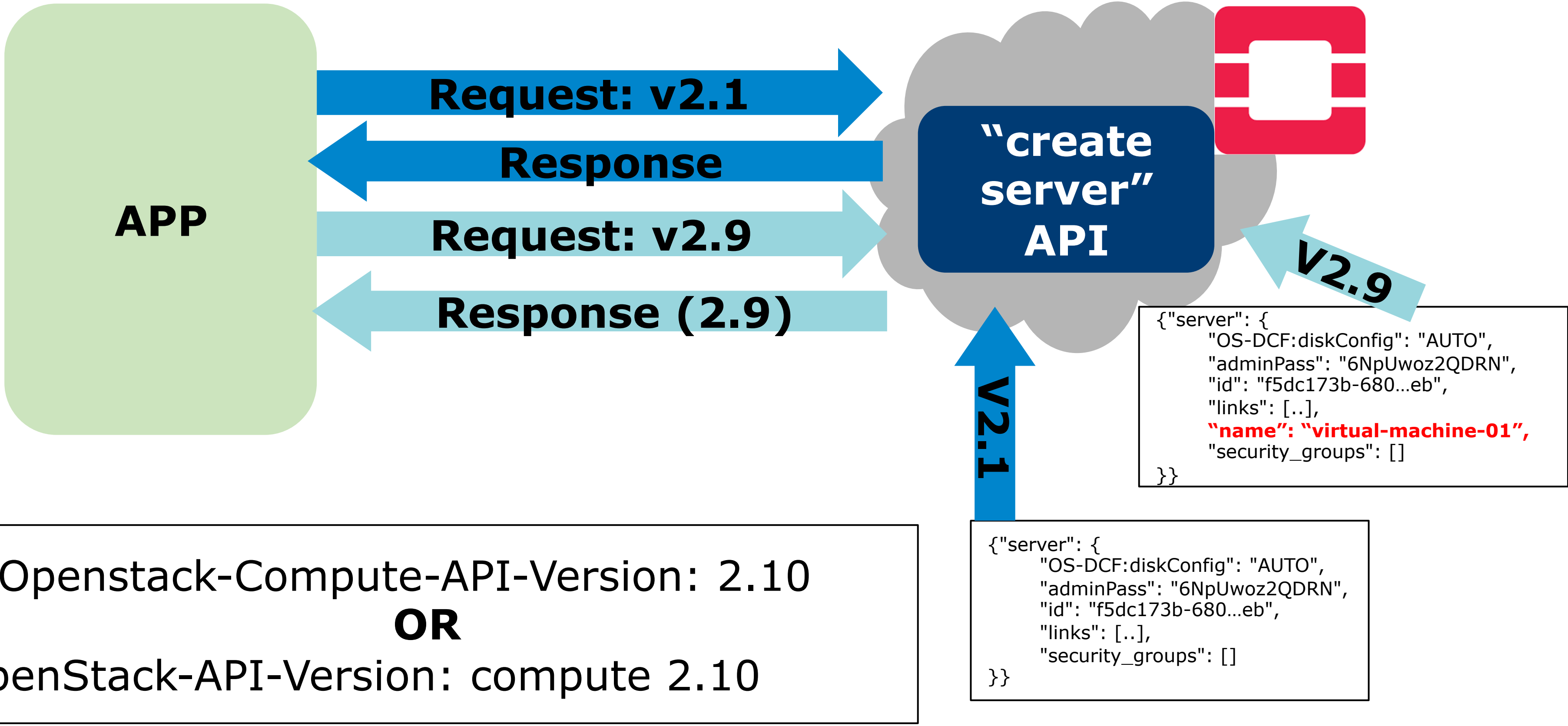
X-Openstack-Compute-API-Version: 2.10

OR

OpenStack-API-Version: compute 2.10

- OpenStack clouds behave as the specified version.
- If not specifying any version from user side, OpenStack clouds behave as the oldest version for existing users.

OpenStack API Microversions



```
{"server": {
       "OS-DCF:diskConfig": "AUTO",
       "adminPass": "6NpUwoz2QDRN",
       "id": "f5dc173b-680…eb",
       "links": [..],
       “name”: “virtual-machine-01”,
       "security_groups": []
}}
```

```
{"server": {
       "OS-DCF:diskConfig": "AUTO",
       "adminPass": "6NpUwoz2QDRN",
       "id": "f5dc173b-680…eb",
       "links": [..],
       "security_groups": []
}}
```

X-Openstack-Compute-API-Version: 2.10

OR

OpenStack-API-Version: compute 2.10

References

OpenAPI

- https://www.openapis.org/

Microversions mechanism

- https://developer.openstack.org/api-guide/compute/microversions.html

OpenStack Powered

- https://www.openstack.org/brand/openstack-powered/

OpenStack SDK

- https://wiki.openstack.org/wiki/SDKs

Thank You

\Orchestrating a brighter world

NEC