Cheops: a framework to geo-distribute micro-service applications at the Edge

Context

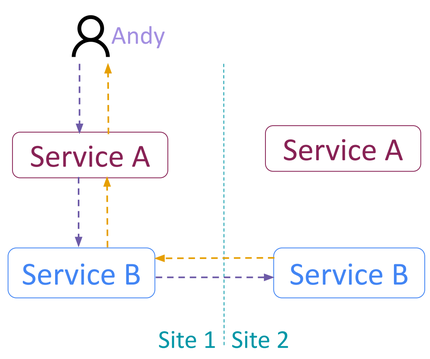
Deployment of micro and nano data centers at the Edge is taking off.

Challenges for applications:

- High latency
- Disconnection between (edge) sites

State-of-the-art

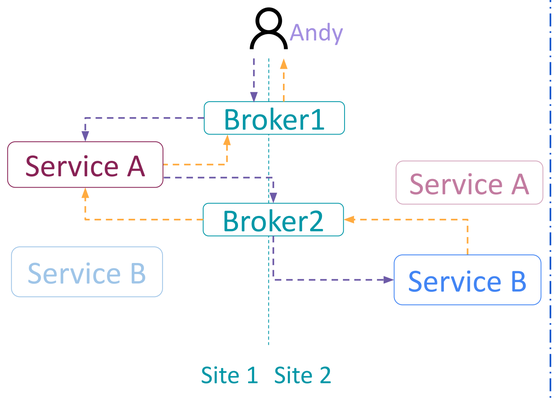
Service to Service Collaboration



Problems:

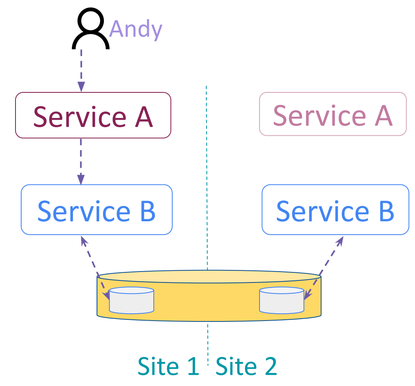
The collaboration code is tangled in the business code

Broker Based Collaboration



Broker needs to be developed for each service

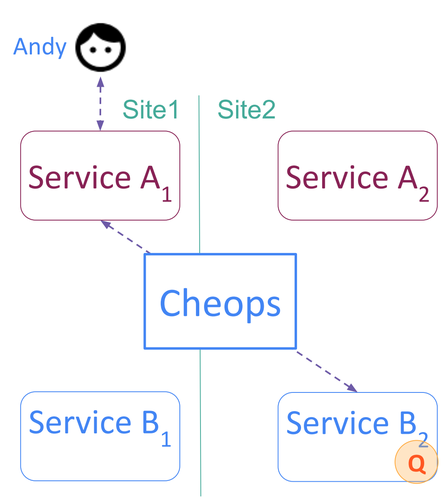
Database Collaboration



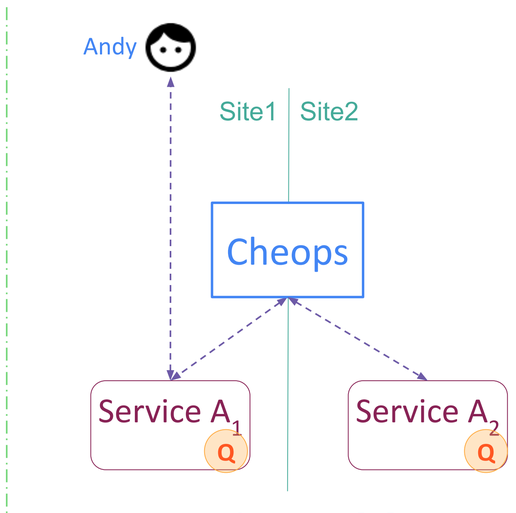
Resources have a context and a scope, sometimes side effects

Principles and collaborations

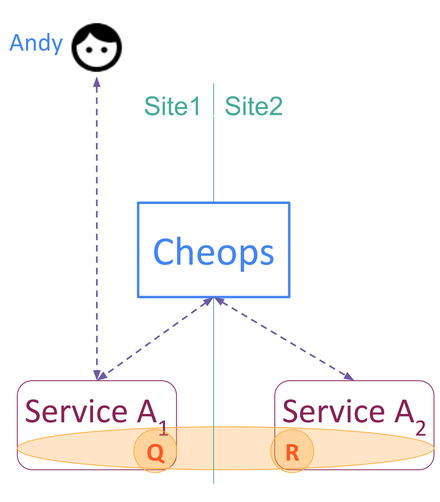
- **Autonomous instances**: local-first for robustness
- **Collaboration** (on demand/if needed): leverage available resources
- **Generic:** the approach should work with multiple applications
- **No touching the code**: no extra efforts (intrusive) to existing code



Sharing:
service B from Site2 has the required resource



Replication[2]:
Andy creates identical resources on different sites



Cross:
Andy creates a resource that span on every involved sites

DSL and Classification

Scope-lang[1] (DSL to manage resources):

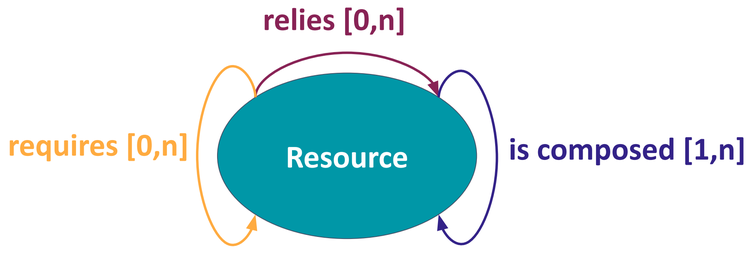
- Expression:
  - Native to CLI
  - Service & location information

```
application serviceA create --sub-resourceB foo
--scope {ServiceA: Site1, ServiceB: Site2}
```

- openstack server create [...] myvm --scope {Nova: Amiens & Nantes}
- Kubectl apply -f nginx.yml --scope {Amiens & Nantes)

A resource is not just a black box!

How to deal with dependencies for a geo-distributed application?

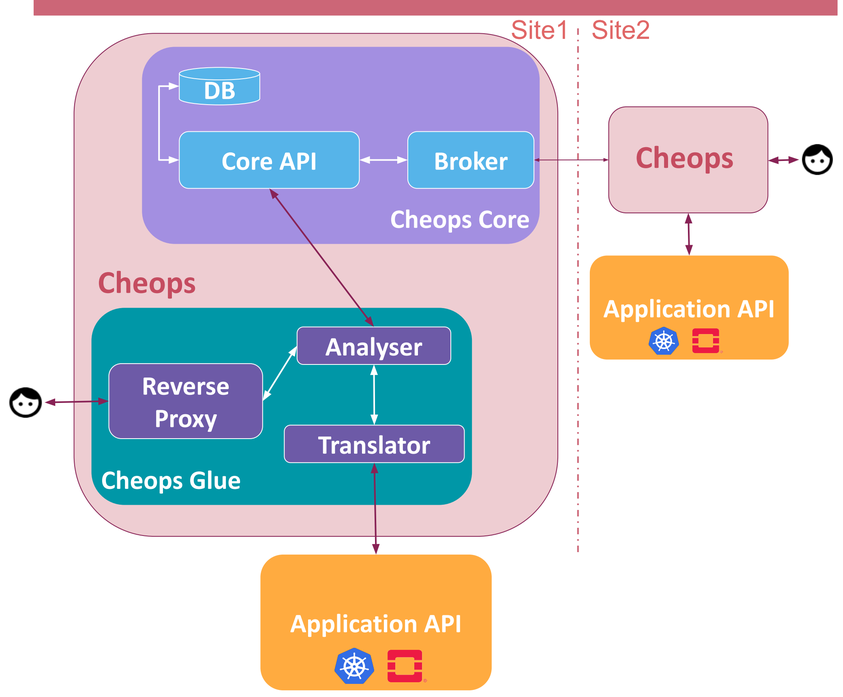


Example:
A vm **requires** an image

Example:
A vm **relies** on an IP

Example:
A deployment **is composed of** pods

Architecture



Cheops agents talk to each other through the broker

Future work

- Autonomous loops similar to a Kubernetes CRD
- Autonomous site optimisation
- Better network partition handling
- More development on Scope-lang & Cheops Framework

Project repository:
https://gitlab.inria.fr/discovery/cheops

[1] Geo-Distribute Cloud Applications at the Edge https://hal.inria.fr/hal-03212421/
[2] A service mesh for collaboration between geo-distributed services: the replication case https://hal.inria.fr/hal-03282425

Geo Johns Antony, geo-johns.antony@inria.fr
Marie Delavergne, marie.delavergne@inria.fr

COLORS
COLORBLIND

Cheops: a framework to geo-distribute micro-service applications at the Edge

Context

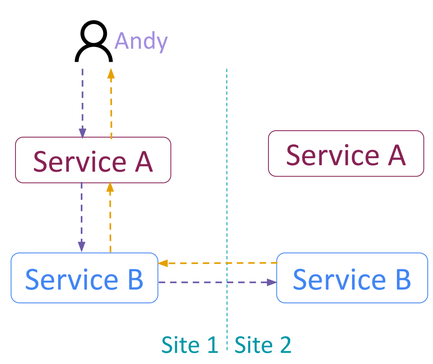
Deployment of micro and nano data centers at the Edge is taking off.

Challenges for applications:

- High latency
- Disconnection between (edge) sites

State-of-the-art

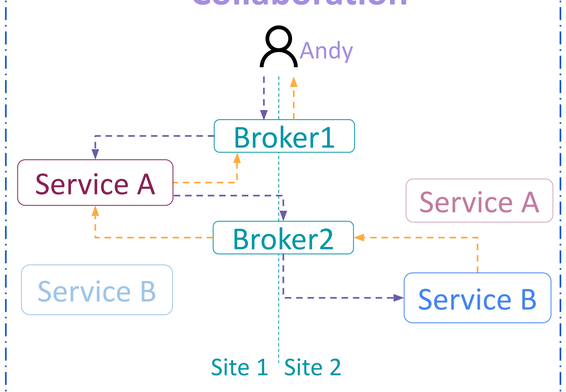
Service to Service Collaboration



Problems:

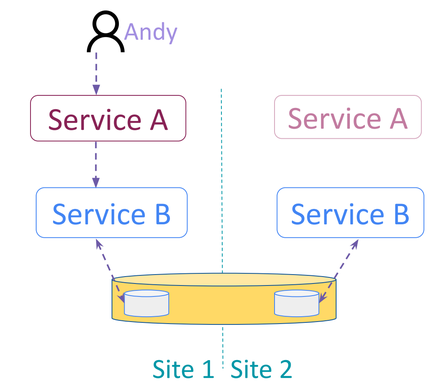
The collaboration code is tangled in the business code

Broker Based Collaboration



Broker needs to be developed for each service

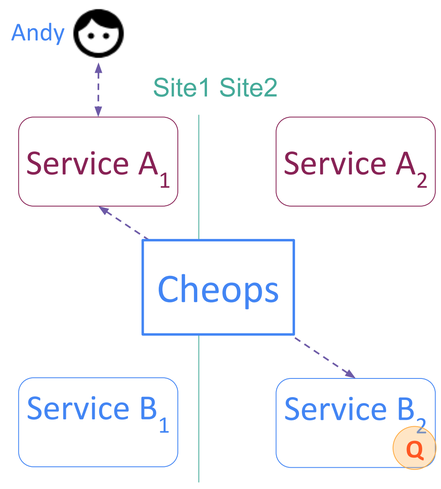
Database Collaboration



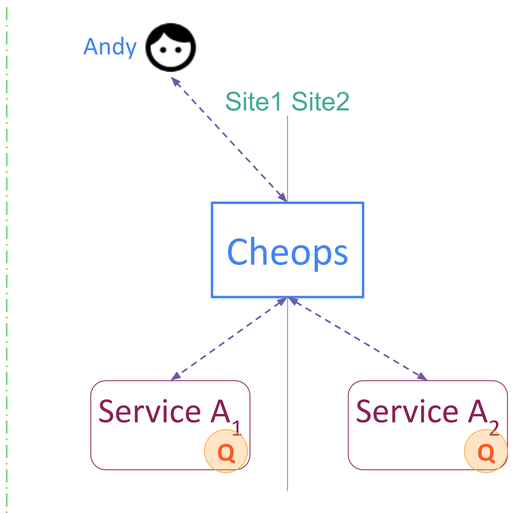
Resources have a context and a scope, sometimes side effects

Principles and collaborations

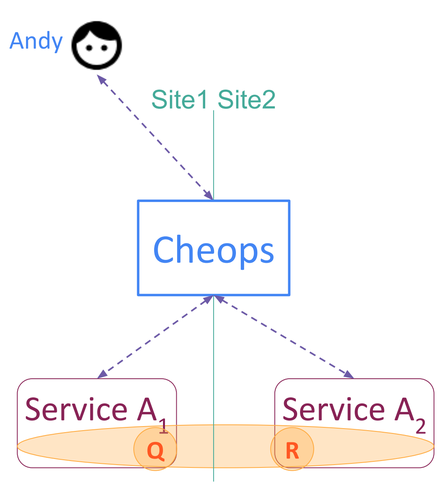
- **Autonomous instances**: local-first for robustness
- **Collaboration** (on demand/if needed): leverage available resources
- **Generic:** the approach should work with multiple applications
- **No touching the code**: no extra efforts (intrusive) to existing code



Sharing:
service B from Site2 has the required resource



Replication:
Andy creates identical resources on different sites



Cross:
Andy creates a resource that span on every involved sites

DSL and Classification

Scope-lang[1] (DSL to for resource management):

- A DSL to manage resources
- Integrated with native CLI
- Irrespective to any platform
- A scope-lang expression contains:
  - Collaboration
  - Service information
  - Location Information

```
application serviceA create --sub-resourceB foo
--scope {ServiceA: Site1, ServiceB: Site2}
```

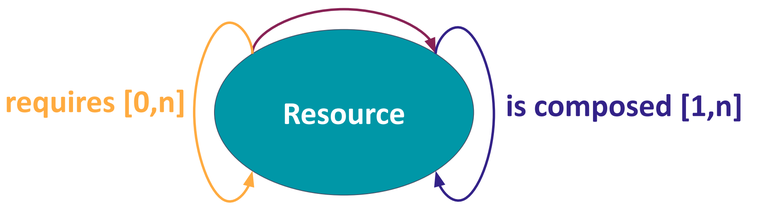
```
openstack server create [...] myvm --scope {Nova:
Berlin & Paris}
```

```
kubectl apply -f nginx.yml --scope {Berlin & Paris}
```

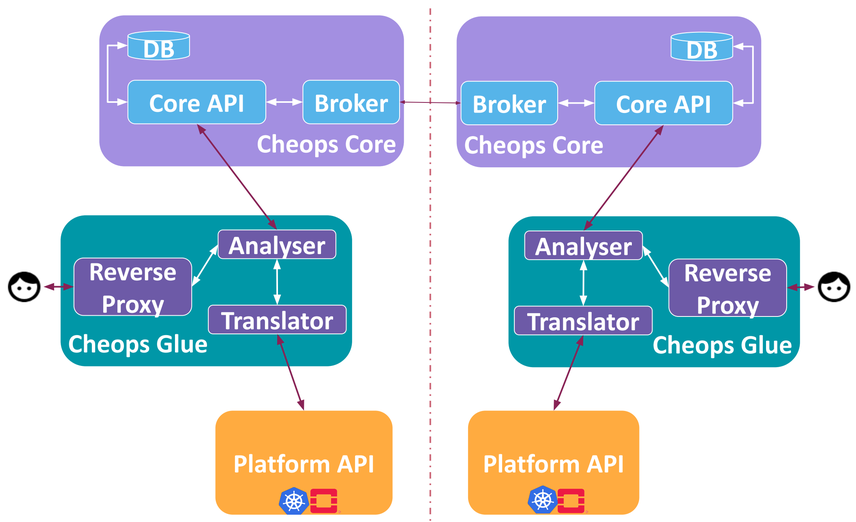
Example:
A vm **requires** an image

Example:
A vm **relies** an IP

Example:
A deployment **is composed of** pods



Architecture



Future work

- Autonomous loops similar to a Kubernetes CRD
- Autonomous site optimisation
- Better network partition handling
- More development on Scope-lang & Cheops Framework

[1] https://hal.inria.fr/hal-03212421/

Geo Johns Antony, geo-johns.antony@inria.fr
Marie Delavergne, marie.delavergne@inria.fr