

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

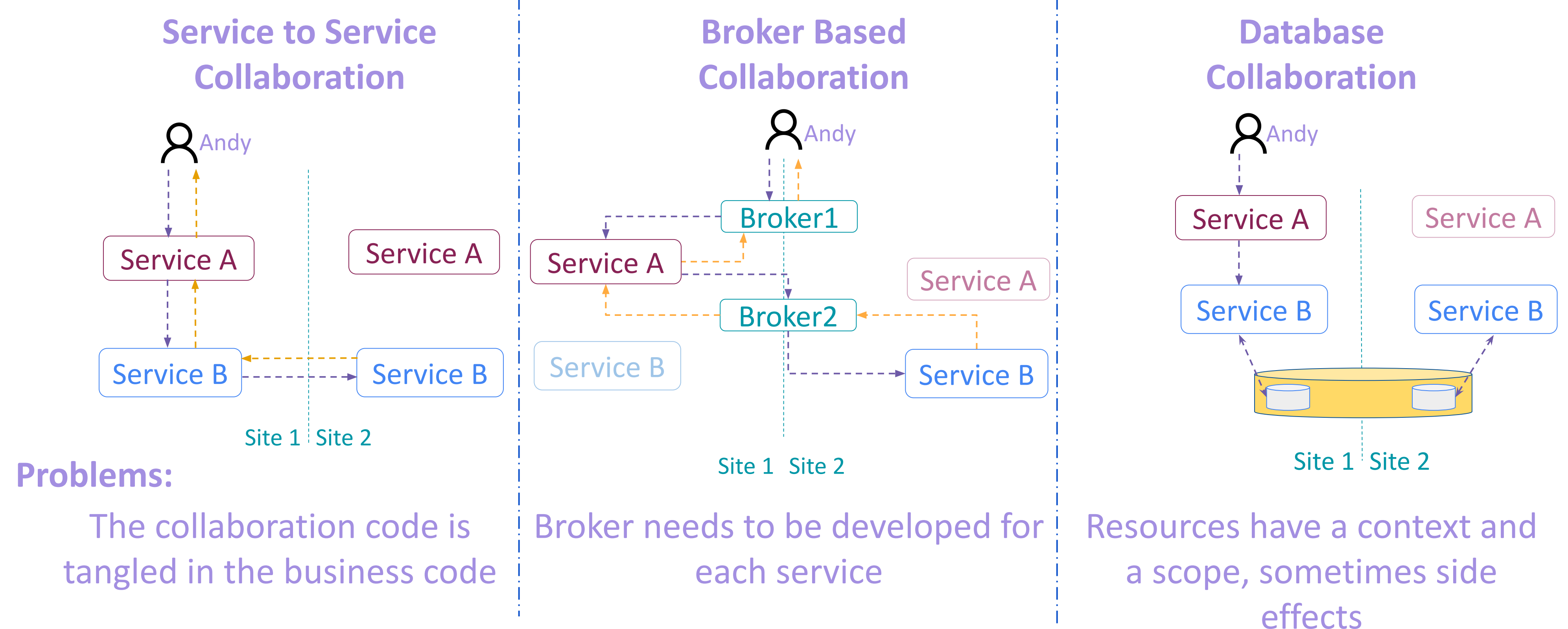
## Context

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

### Challenges for applications:

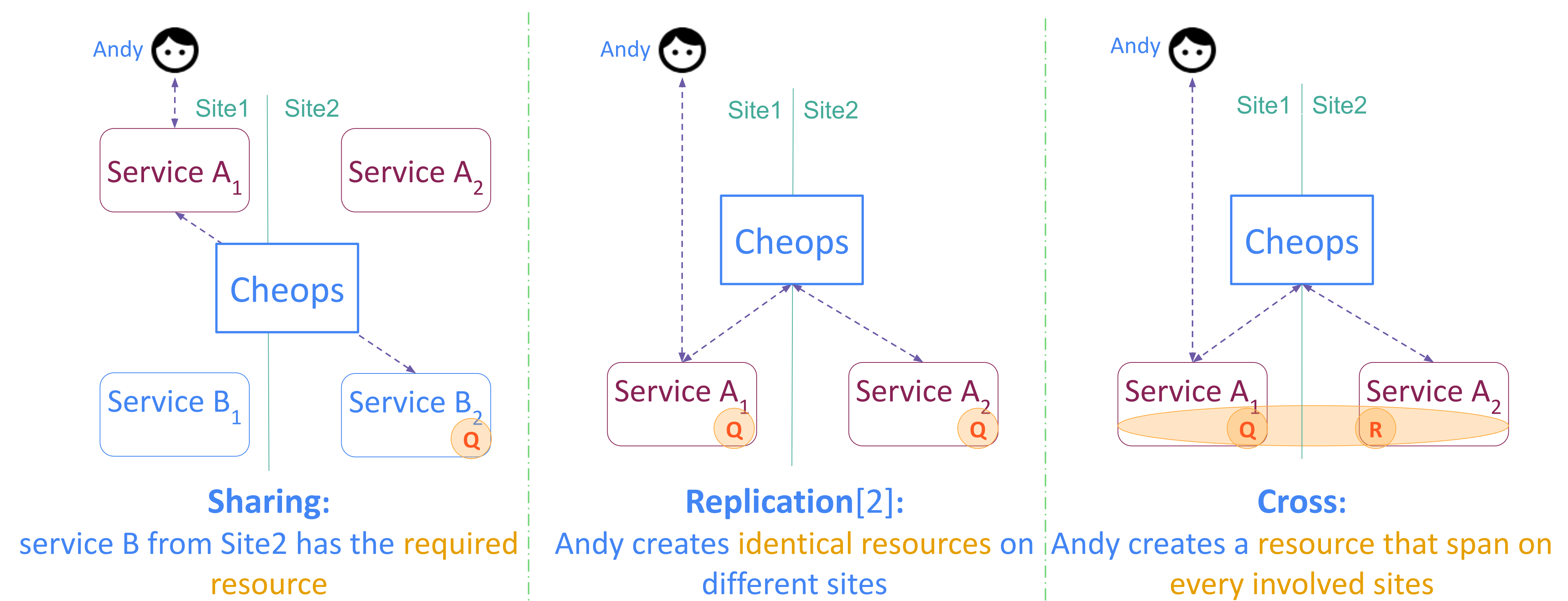
- High latency
- Disconnection between (edge) sites

## State-of-the-art



## 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



## DSL and Classification

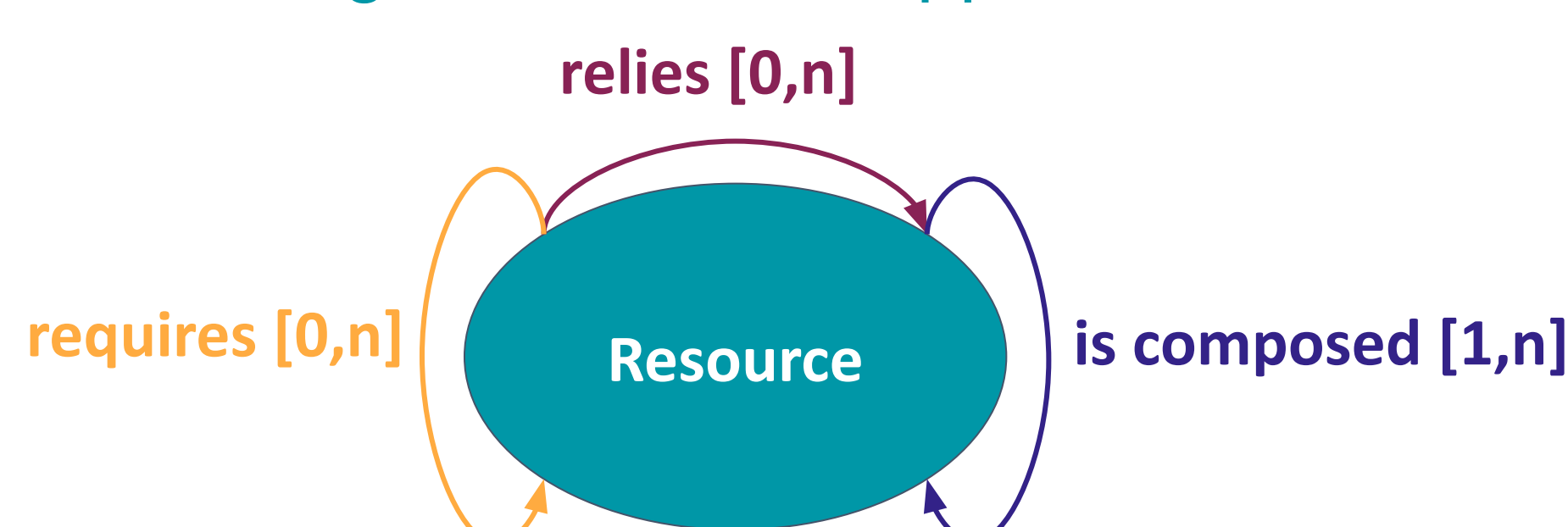
Scope-lang<sup>[1]</sup> (DSL to manage resources):

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

```
application serviceA create --sub-resourceB foo
--scope {ServiceA: Site1, ServiceB: Site2}
o openstack server create [...] myvm --scope {Nova: Amiens & Nantes}
o 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?

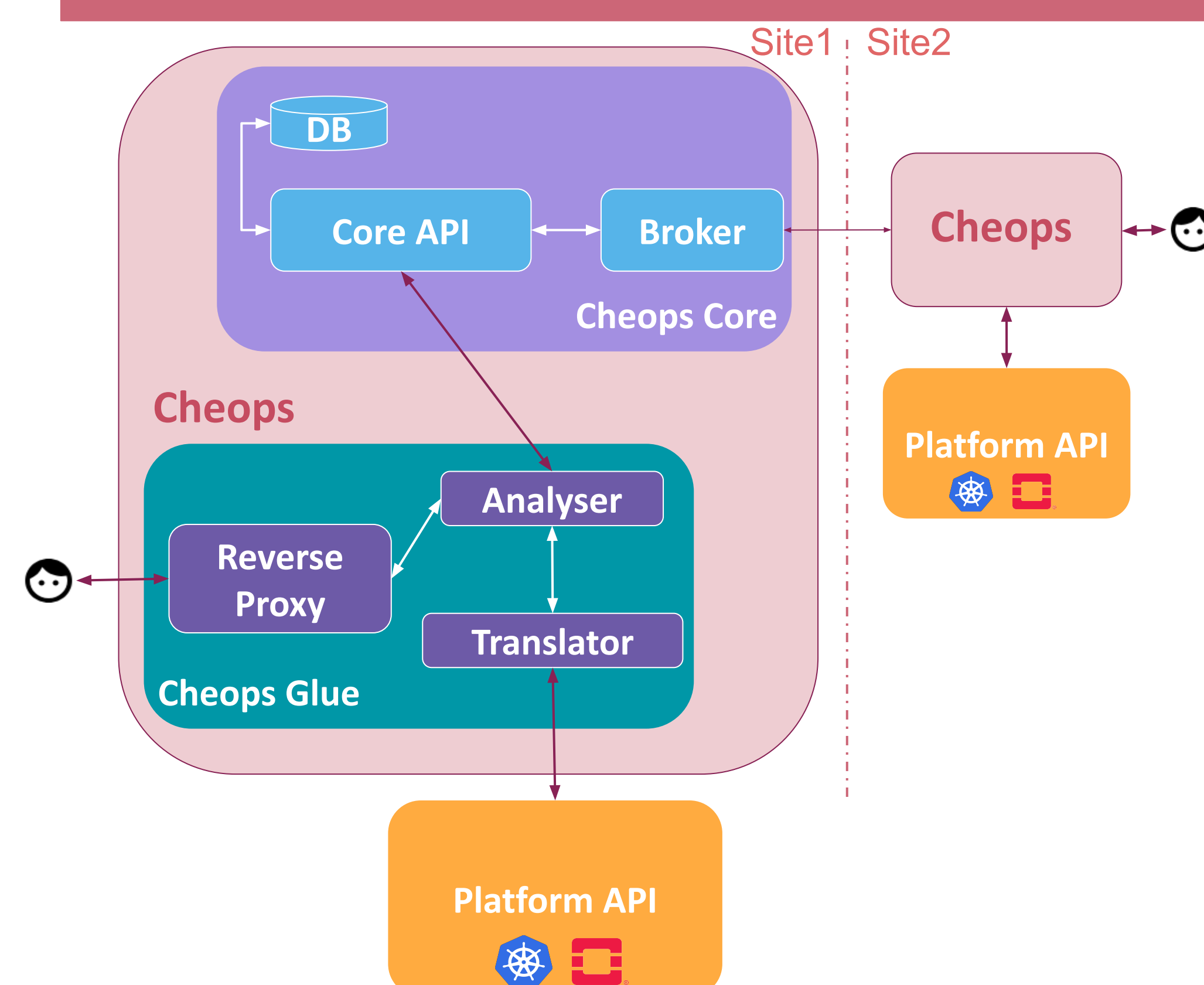


Example:  
A vm **requires** an image

Example:  
A vm **relies** 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>

