

---

# ProActive and Fractal

Matthieu Morel, Denis Caromel

OASIS Team

INRIA Sophia - CNRS - I3S

---

---

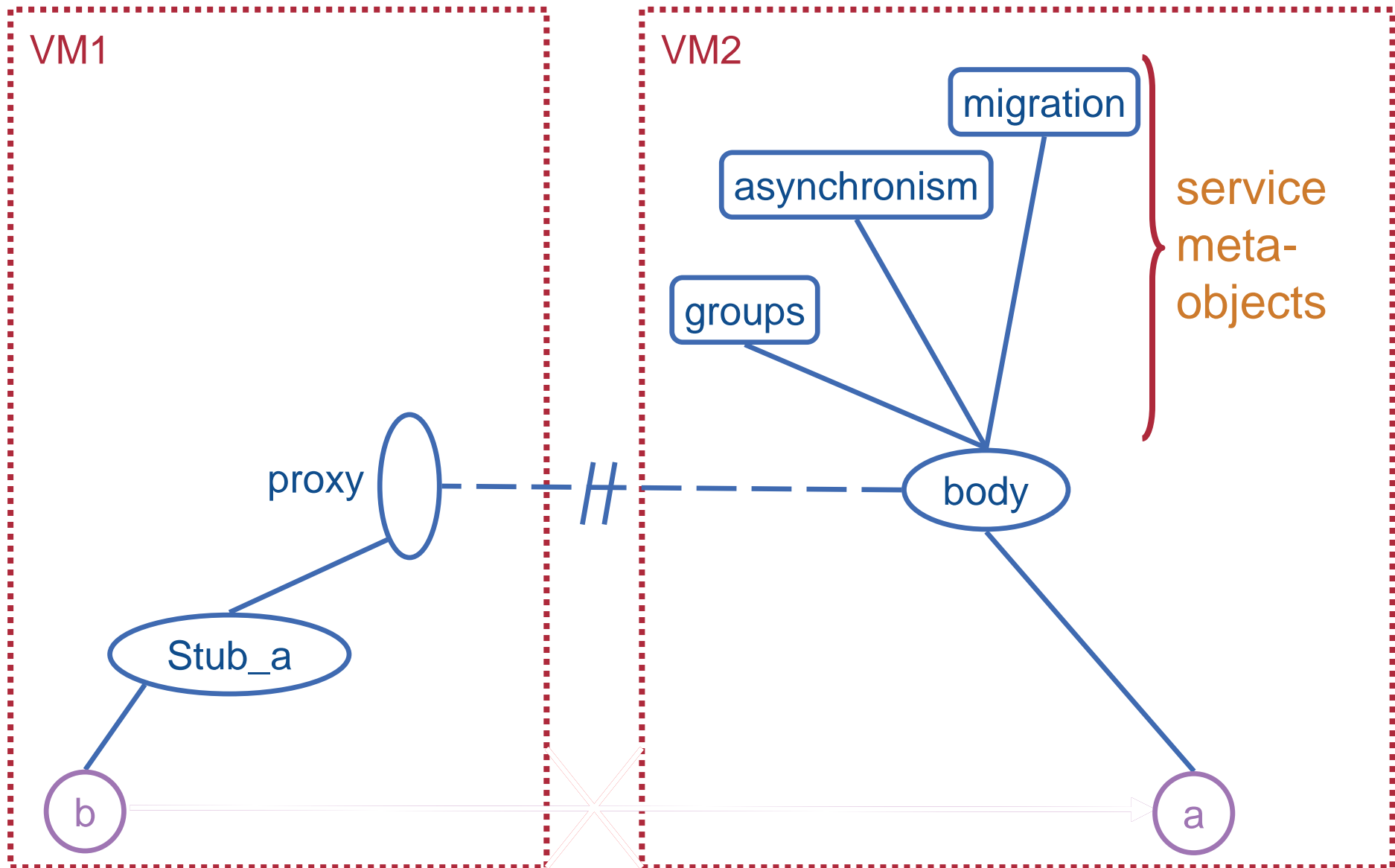
# Working context

- ❑ ProActive :

  - a library in java for **parallel, distributed** and **concurrent** computing with **mobility** and **security**

- ❑ Active Objects





---

# Working context

## ❑ ProActive :

a library in java for **parallel, distributed** and **concurrent** computing with **mobility** and **security**

## ❑ Active Objects

## ❑ usage

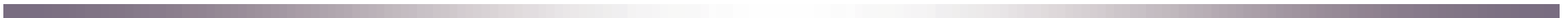
```
A a = new A(params);
```



```
A a = (A) ProActive.newActive(« A », params, node);
```



```
ComponentIdentity c = ProActive.newActiveComponent(  
    « A », componentParams, node);
```



---

# A component model for ProActive?

## □ ARCAD

### □ Purposes

- ◆ distributed (and parallel) components
- ◆ hierarchies (high level views on subsystems)
- ◆ dynamic and interactive creation,  
assembly and binding

- ◆ ease of deployment
- ◆ on GRIDs
- ◆ mobility

} entities

} infrastructure

---

---

# Approach

## □ Julia

- ◆ not distributed
- ◆ control objects independent of active objects

one would need to adapt ProActive to Julia!

⇒ Own Fractal implementation

---

---

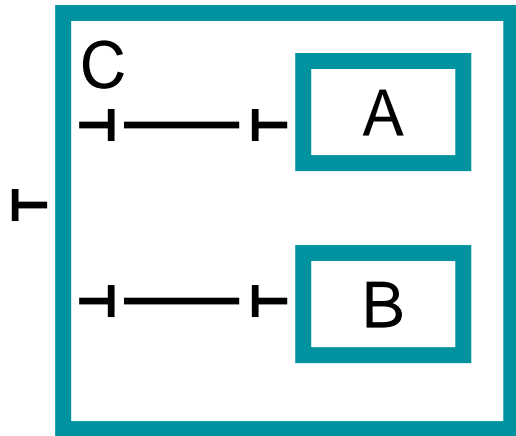
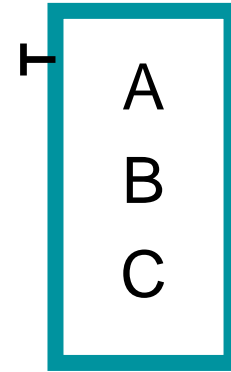
# Fractal implementation : introduction

## □ Preliminary considerations

- ◆ no templates
- ◆ Fractal 1.0.4 : no objectweb.naming
- ◆ no life cycle controllers

## □ ProActive component

- ◆ 1 component = 1 (+) active object
  - ◆ component types
-



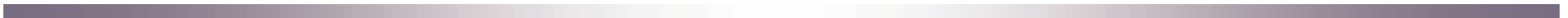


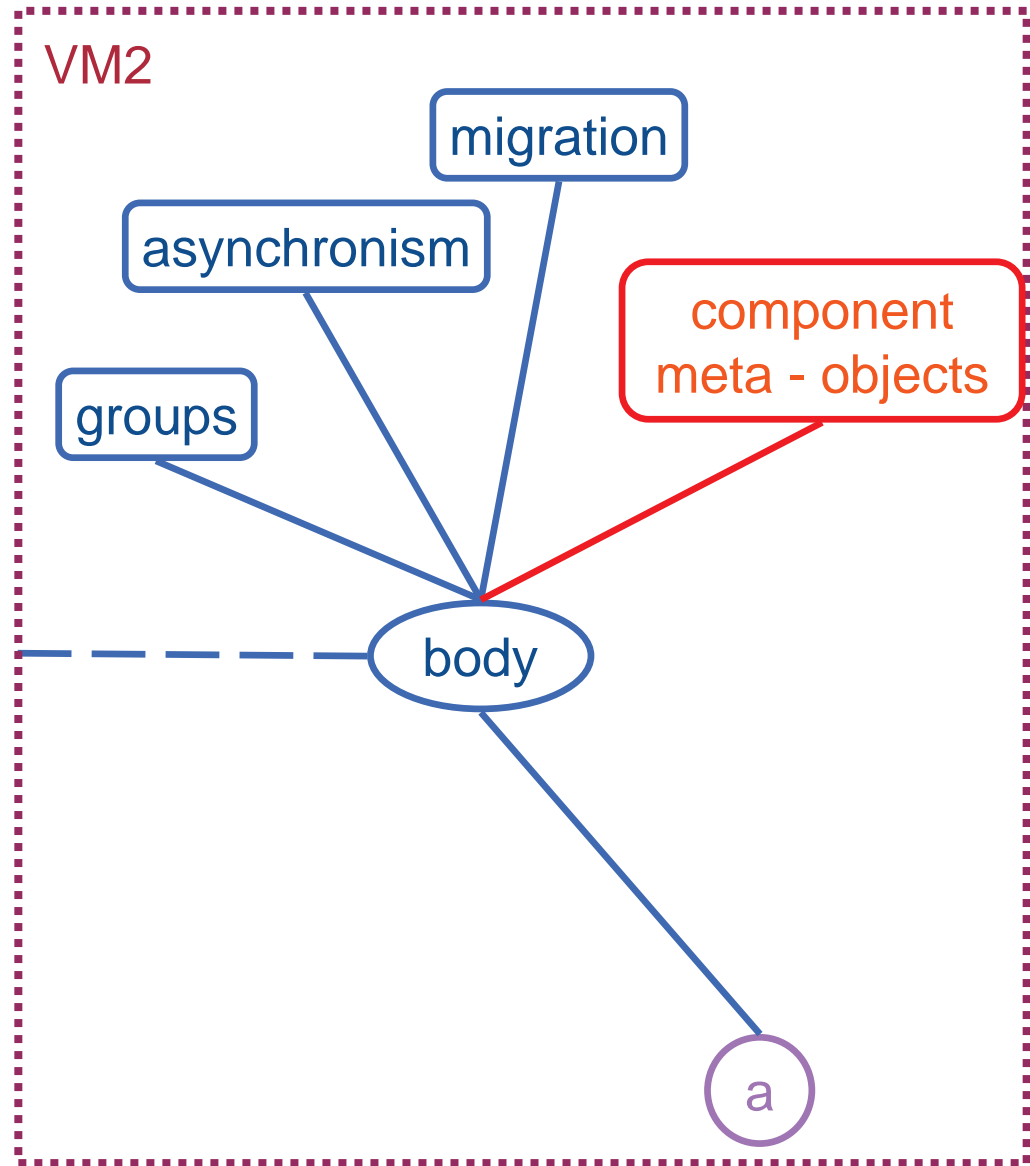
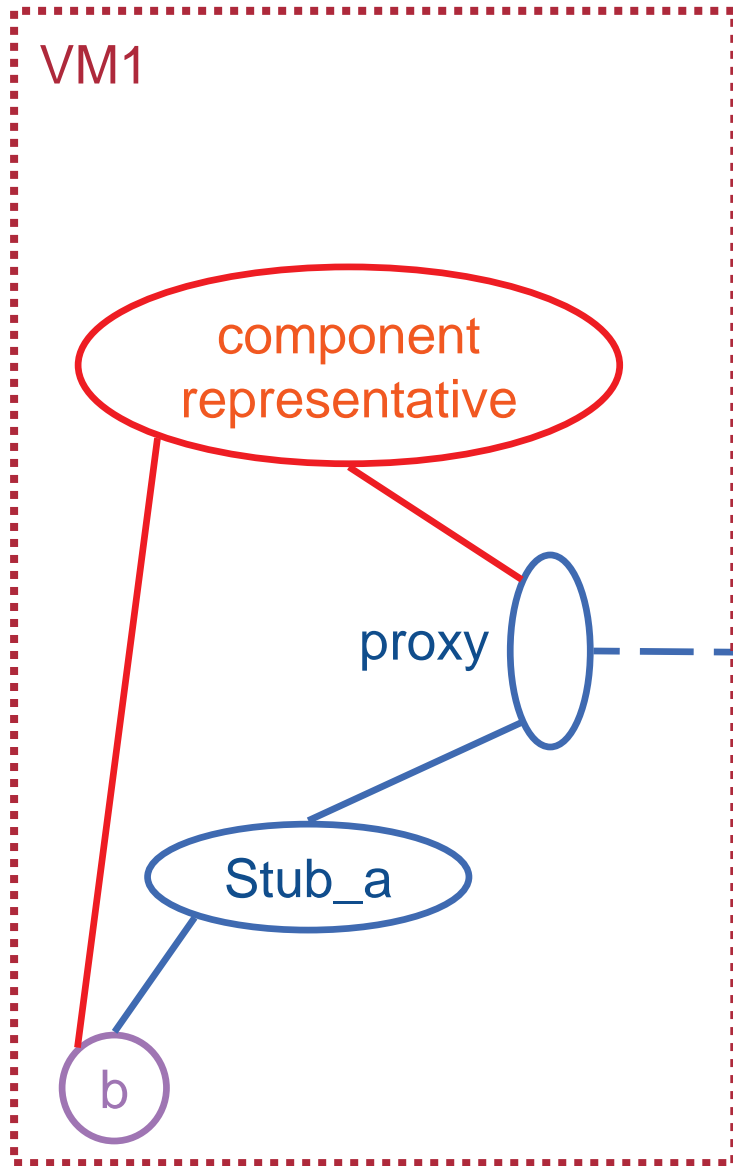
---

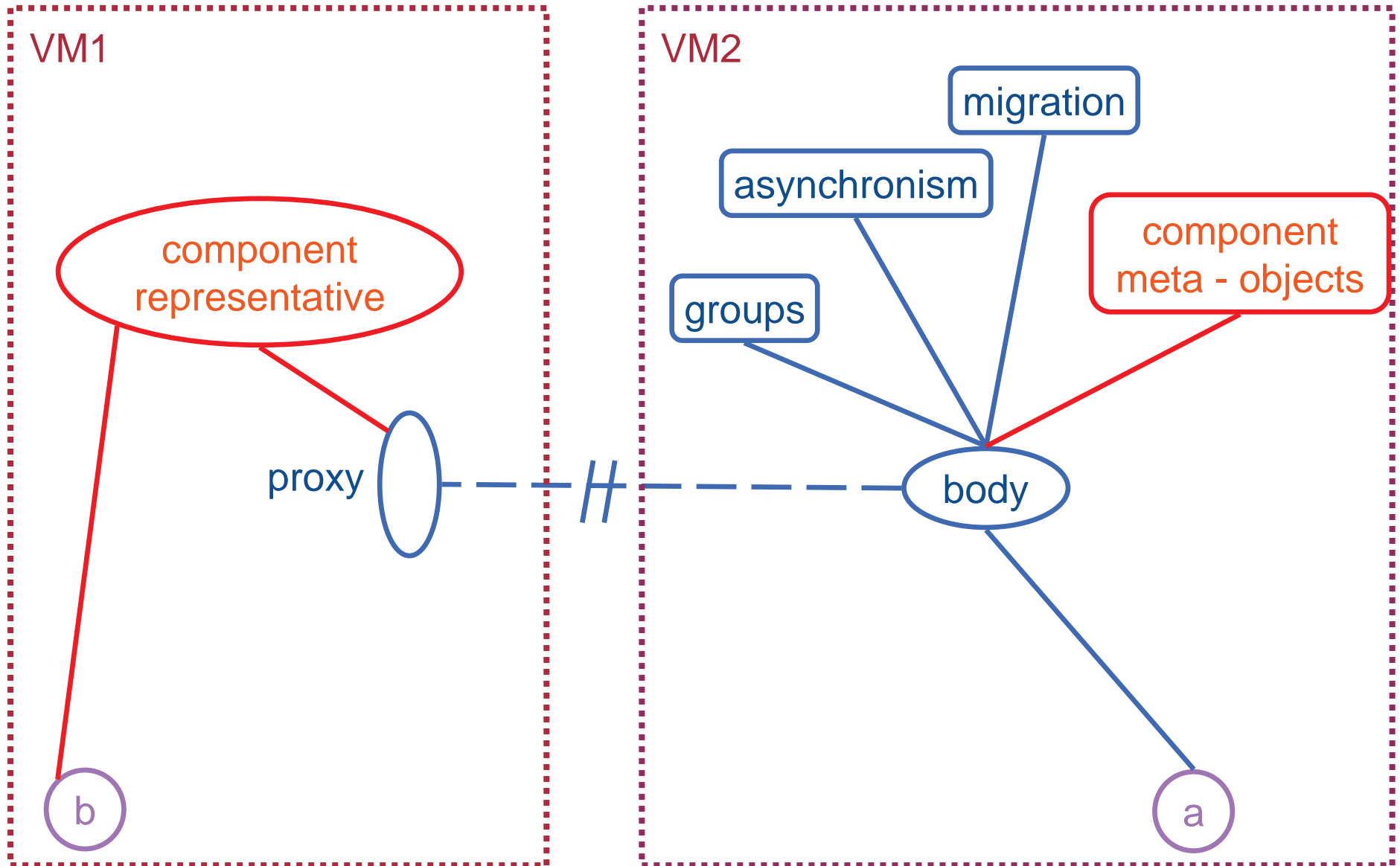
# Fractal implementation : architecture

## □ principles

- ◆ component meta-object
- ◆ component stub (representative)







---

# Fractal implementation : architecture

## □ principles

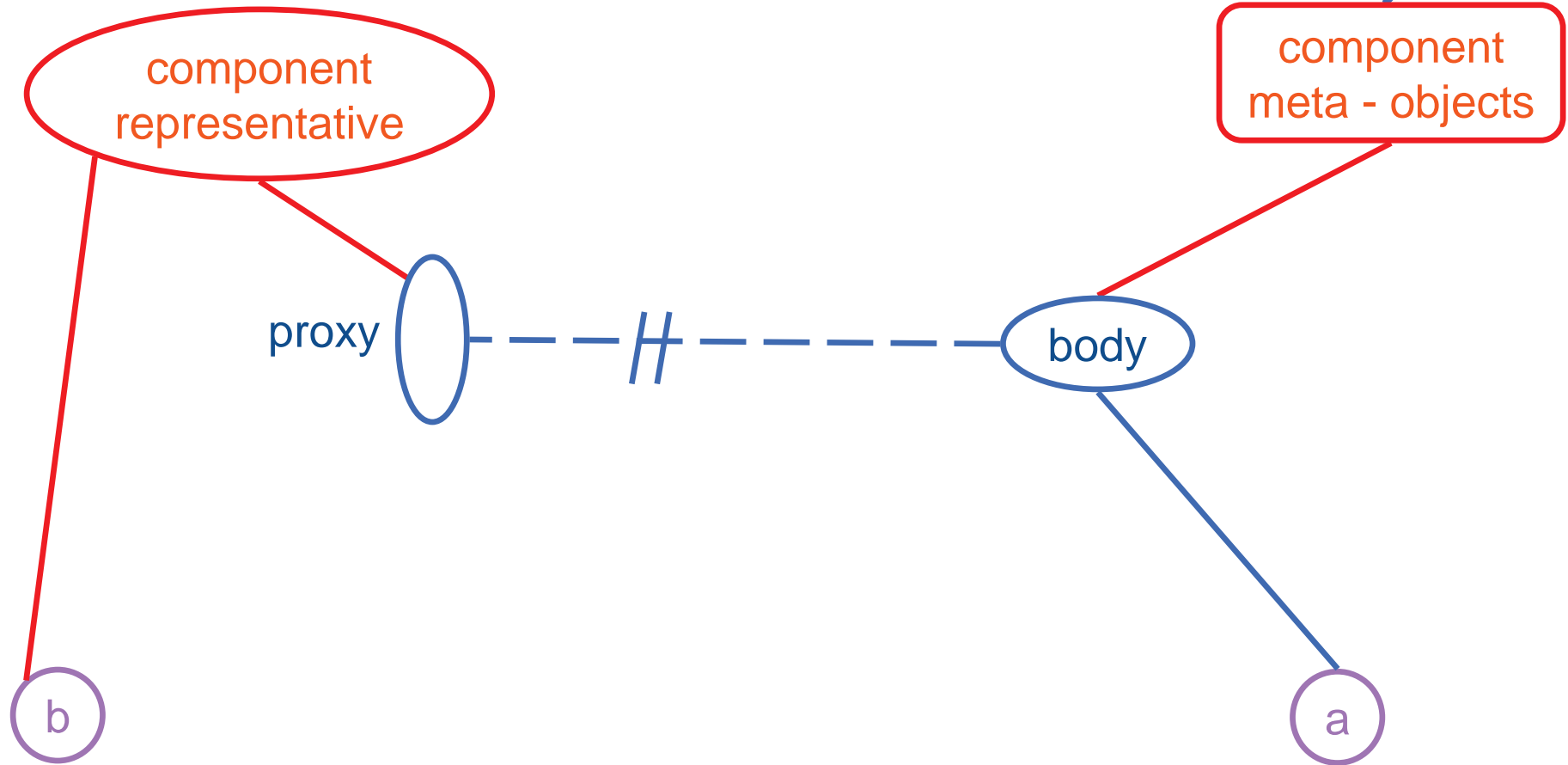
- ◆ component meta-object
- ◆ component stub (representative)
- ◆ Fractal interfaces / functional interface references



---

Fractal Interfaces

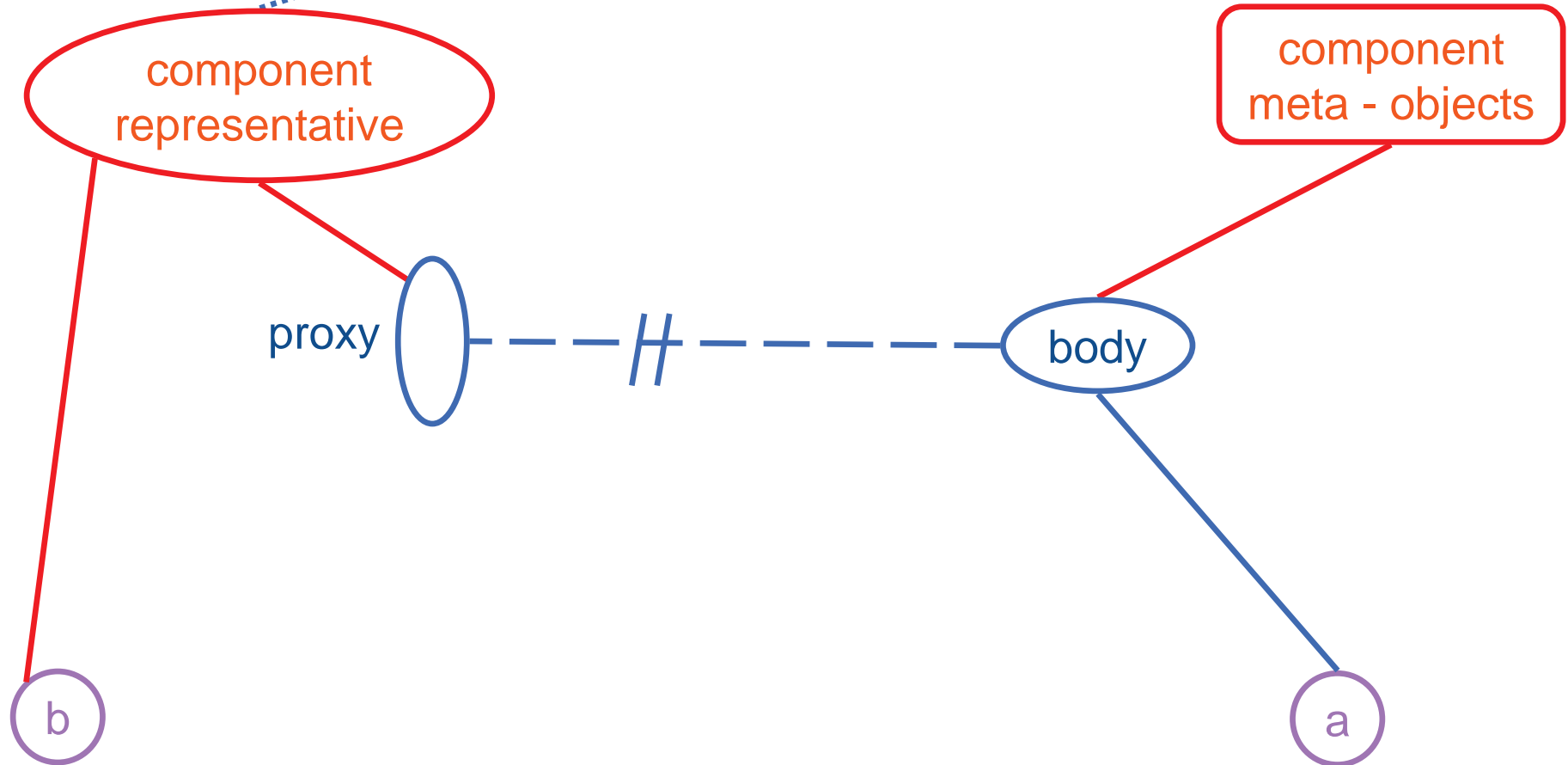
Functional Interfaces



---

Fractal Interfaces

Functional Interfaces



---

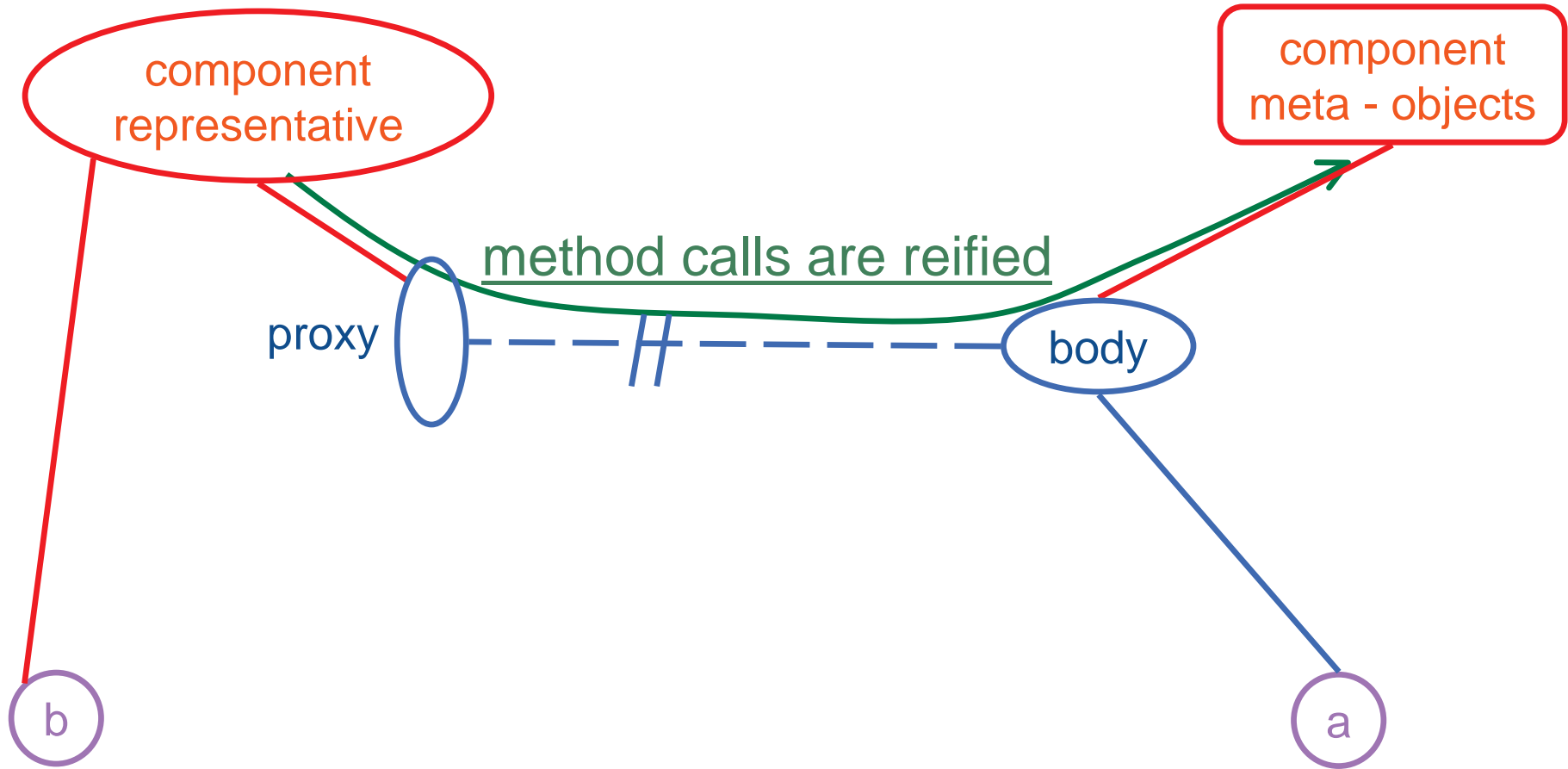
# Fractal implementation : architecture

## □ principles

- ◆ component meta-object
- ◆ component stub (representative)
- ◆ Fractal interfaces / functional interface references

## □ communications

- ◆ reification
-





---

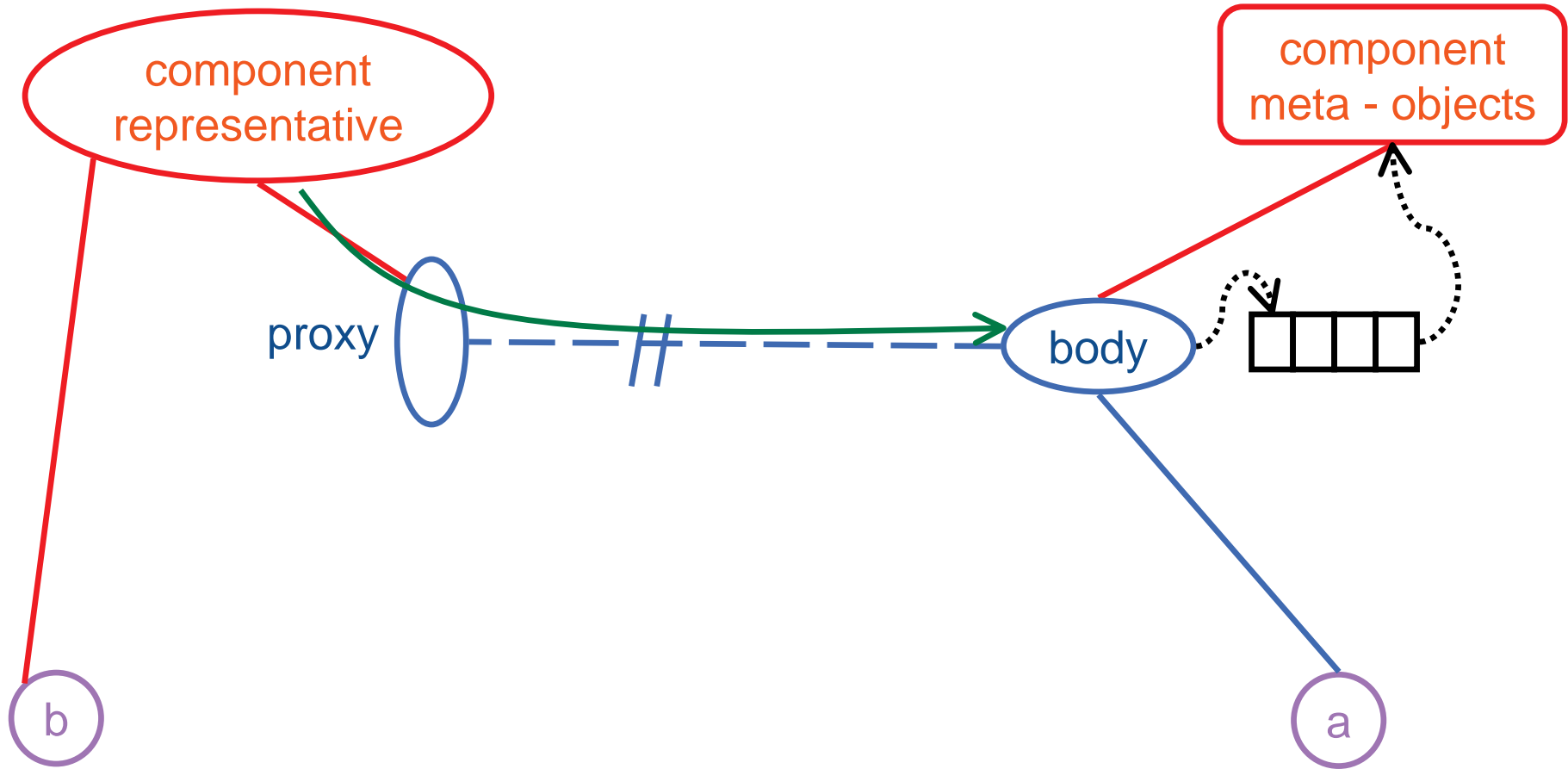
# Fractal implementation : architecture

## □ principles

- ◆ component meta-object
- ◆ component stub (representative)
- ◆ Fractal interfaces / functional interface references

## □ communications

- ◆ reification
  - ◆ request queues (Julia interceptors)
-



---

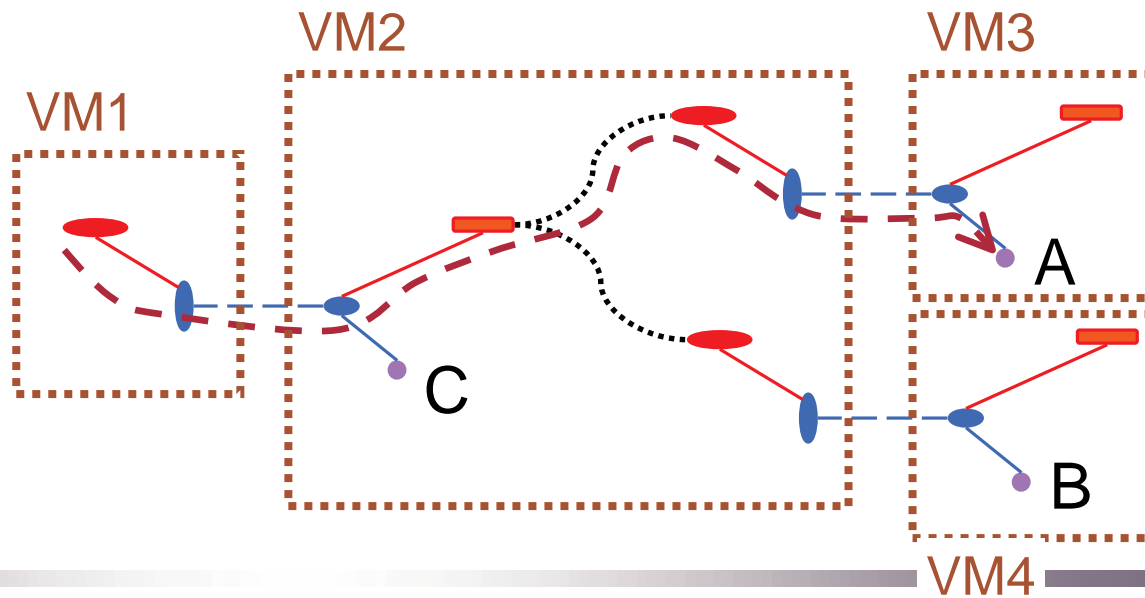
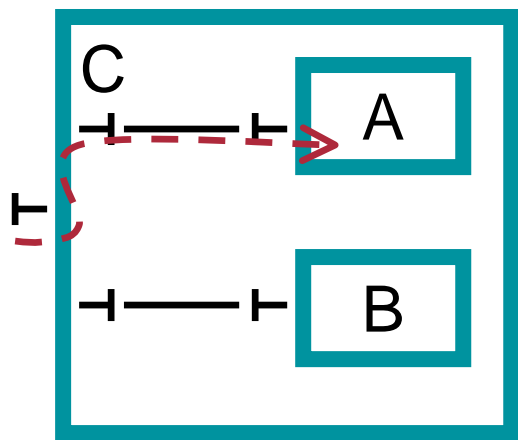
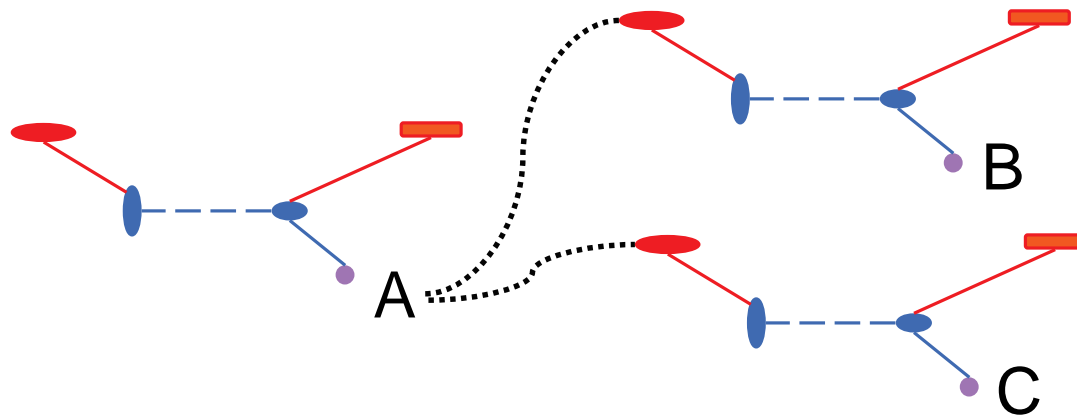
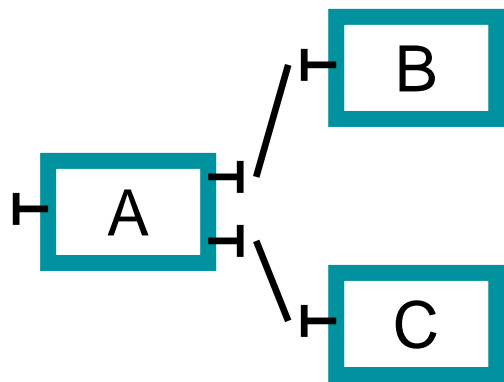
# Fractal implementation : architecture

## □ principles

- ◆ component meta-object
- ◆ component stub (representative)
- ◆ Fractal interfaces / functional interface references

## □ communications

- ◆ reification
  - ◆ request queue (Julia interceptors)
  - ◆ bindings
-



---

# Future work

- ◆ life cycle controllers
  - ◆ group communications
  - ◆ direct communications
  
  - ◆ GUI (IC2D extensions)
  - ◆ C3D : a concrete example
  
  - ◆ component descriptor files
  - ◆ templates
-