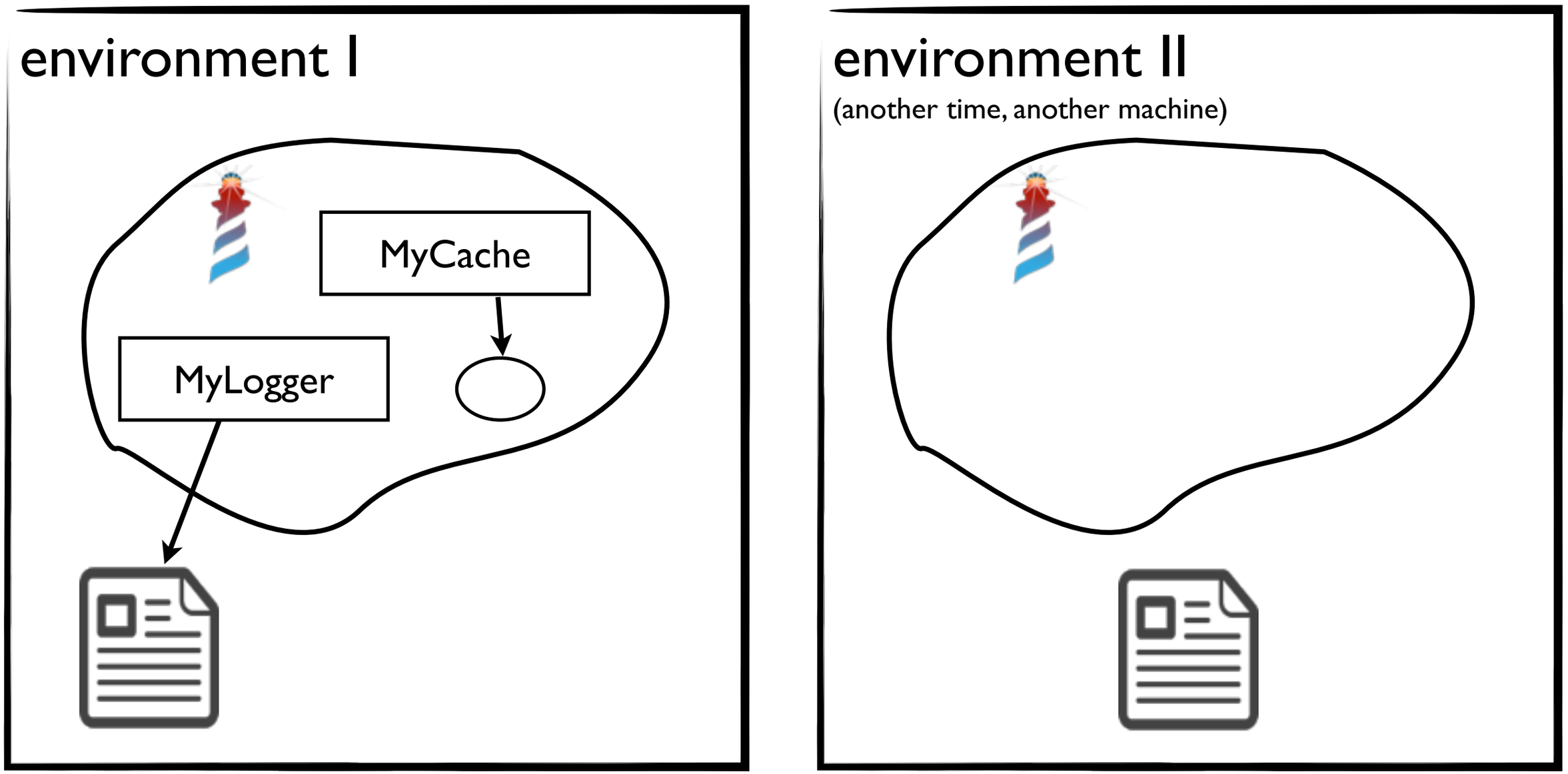
Understanding Pharo's **global state** to **move programs** through **time and space**

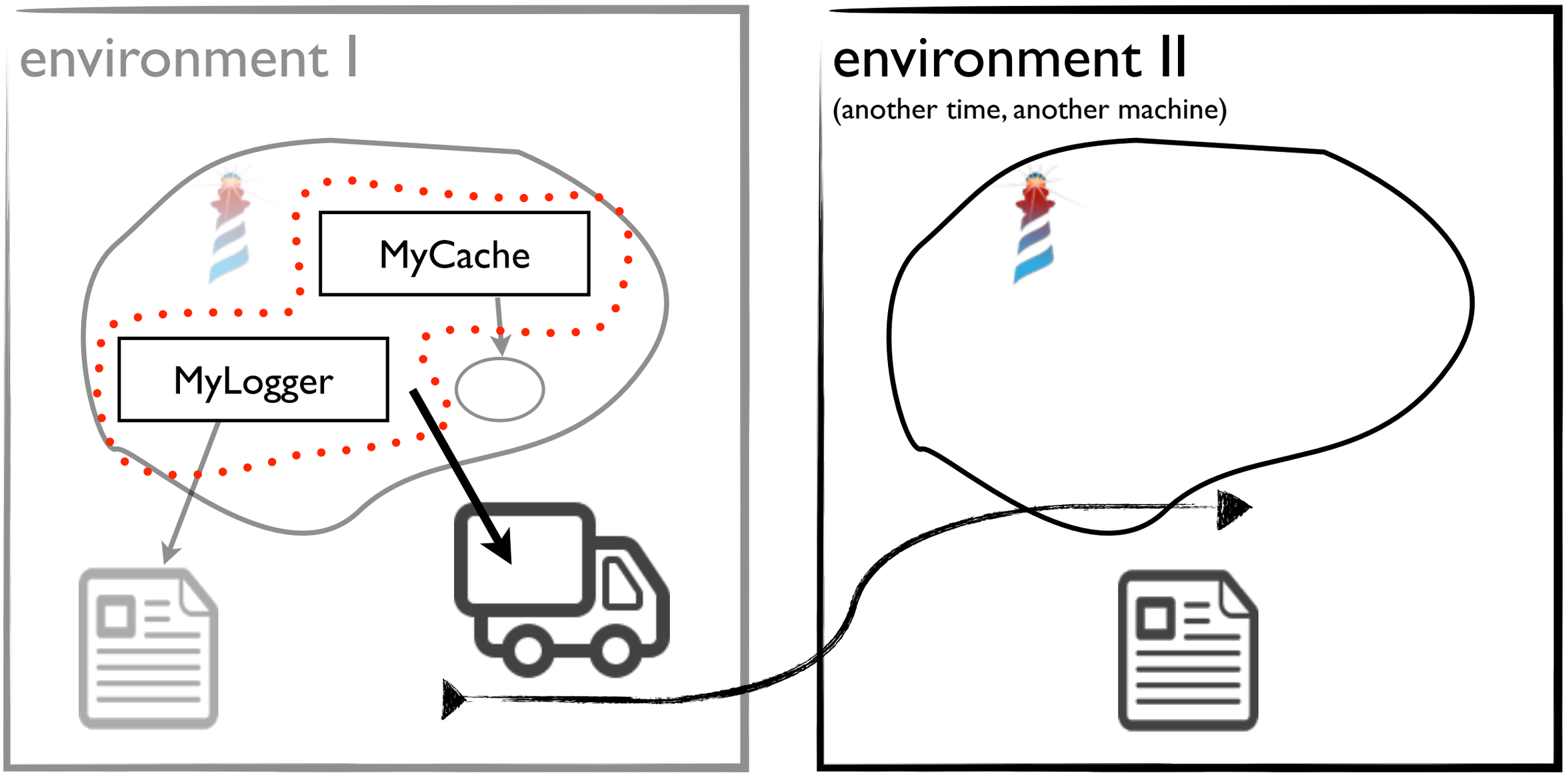
Guillermo Polito, Noury Bouraqadi, Stéphane Ducasse, Luc Fabresse
Ecole des Mines de Douai, RMoD/Inria

Stéphane Ducasse
IWST 2014, ESUG, Cambridge

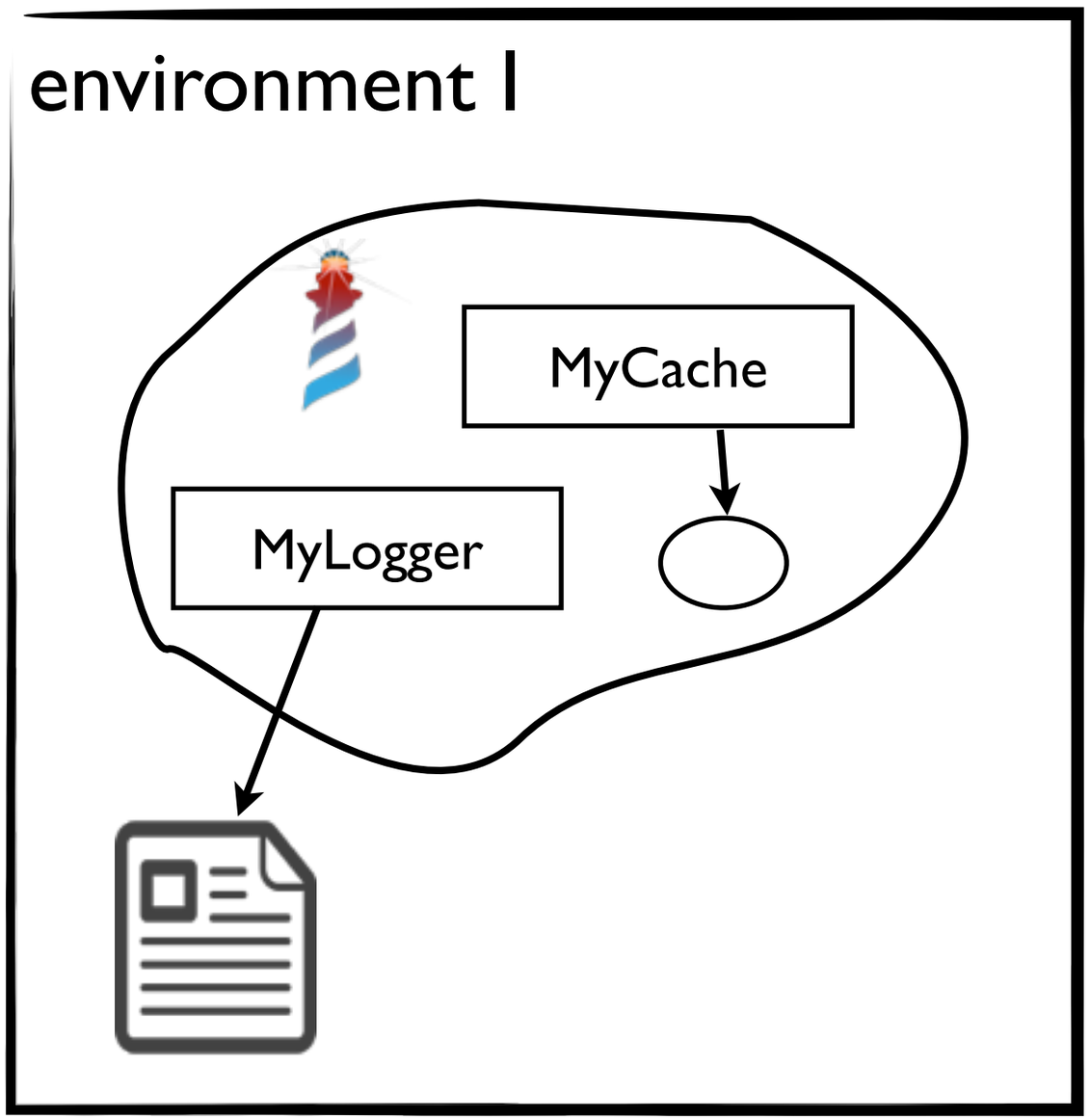
The problem: moving code around

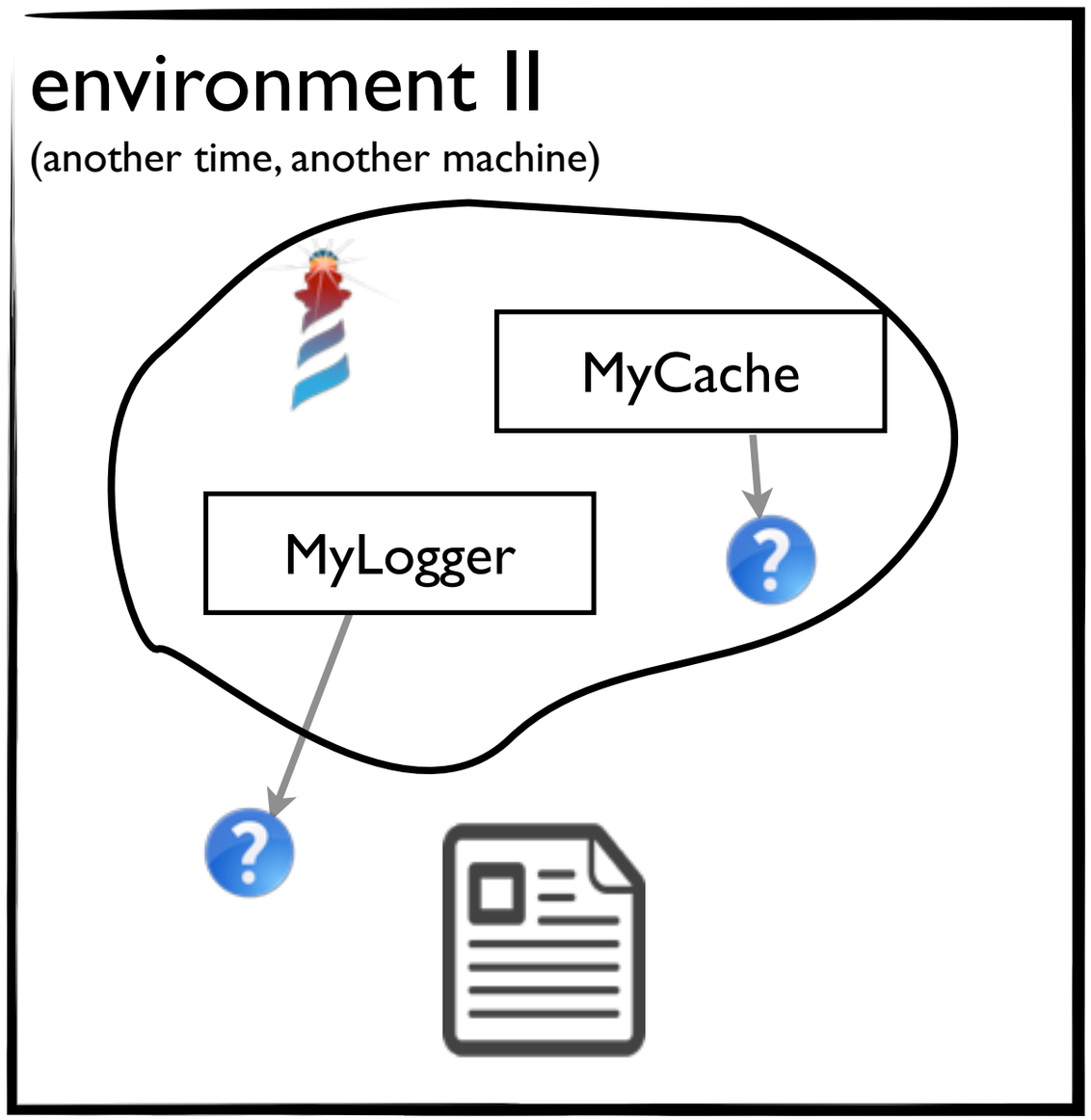


The problem: moving code around



The problem: moving code around





The problem:
moving code with global state around

- should global state be reinitialized?
- should it be kept and moved as well?
- should it be rebound to some already existing object?

The main challenge (by example): non-explicitness

```
WeakIdentityKeyDictionary subclass: #ASTCache
    classVariableNames: 'Default'.

ASTCache>>at: aCompiledMethod
    ^ self
        at: aCompiledMethod
        ifAbsentPut: [ self newASTFor: aCompiledMethod ].

ASTCache>>newASTFor: aMethod
    "creation of the AST..."

ASTCache>>reset
    self removeAll.

ASTCache class>>default
    ^ Default ifNil: [ Default := self new ].

ASTCache class>>shutDown
    self default reset.
```

- Ad-hoc implementation of a cache
  => no clear API
- It's incomplete (e.g., lack support for a recycling strategy)
  => each cache implements its own features
- We know it is a cache just because of the class name
  => no programatic way to identify it

The need:

Understanding

the **global state**

A classification

Category	**#**
Constants	1722
Settings	236
Singletons	65
Graphical Res.	47
Caches	43
Registries	31
Session Specific	27
Process Controllers	11
Finalizables	6

Some possible solutions

- Reification of implicit elements
- Moving responsibilities to the language

Reification provides with

- Explicitness
- Unified APIs
- Frameworks and libraries can profit from it (particularly code-migration ones)

E.g.,
first class instance variables,
first class processes,
first class caches

Moving responsibilities to the language

- So the system can introspect itself
- Modify itself
- And *control* itself

E.g.,
flush caches on low memory

Understanding Pharo's **global state** to **move programs** through **time and space**

Guillermo Polito, Noury Bouraqadi, Stéphane Ducasse, Luc Fabresse
Ecole des Mines de Douai, RMoD/Inria

- A classification of global state usage
- Detected a need for reification
- Detected a need for moving some responsibilities to the language