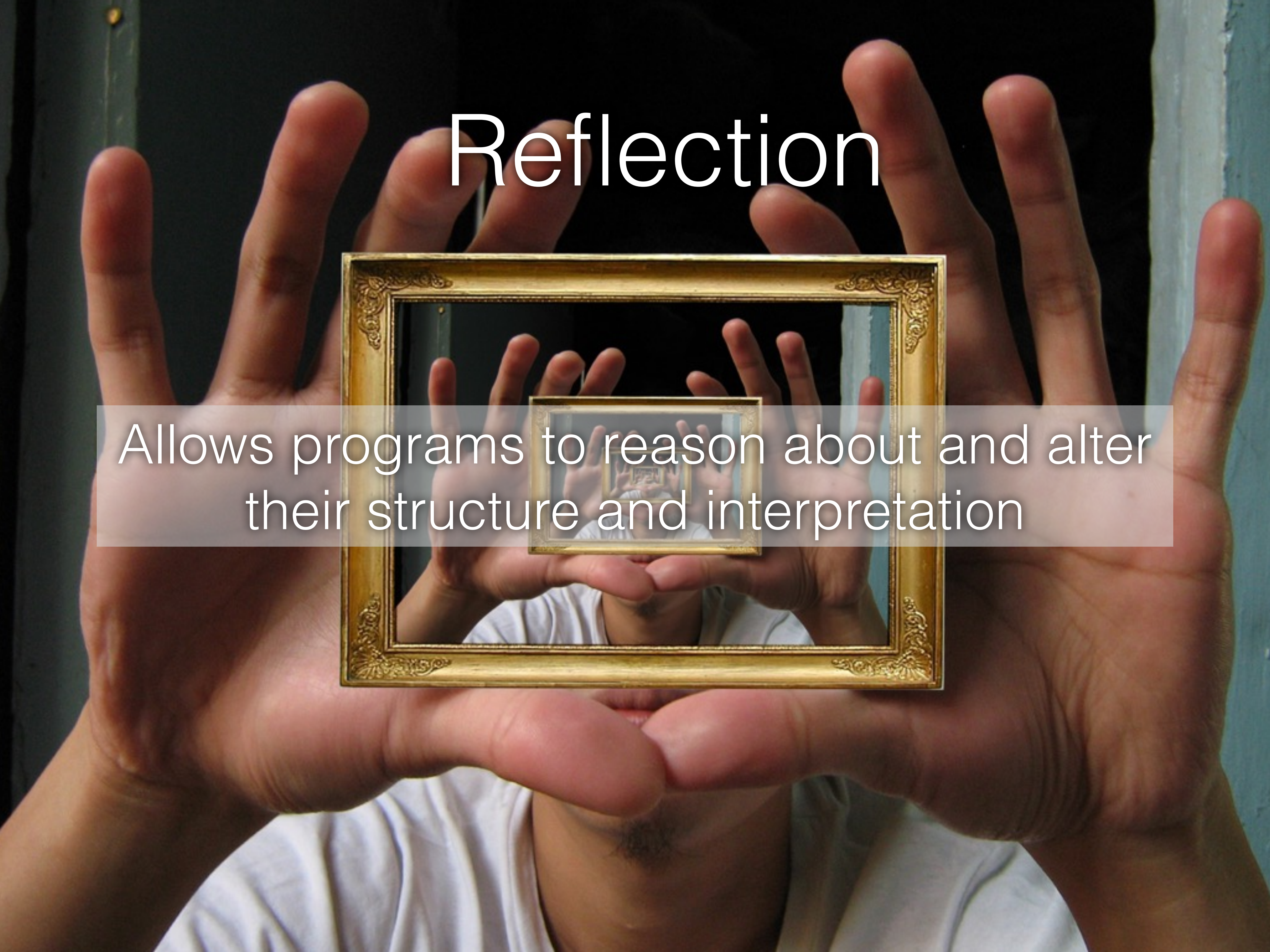
Secure Reflection

Camille Teruel

Reflection

Reflection



Allows programs to reason about and alter their structure and interpretation

Reflection:

Dev tools

Debuggers

Inspectors

Browsers

Dynamic analyses

Profilers

Reflection:

Powerful creative mean

Frameworks

Metaprogramming

Generic programs

Language extensions

Reflection:

Infrastructure

Dynamic Software Updates

Self-Adaptive Programs

Remote-Debugging

Encapsulation ?

Reflection:

Encapsulation’s enemy

Cannot keep things private

From a **modularity** POV : Potential …

… mess!!

Cannot keep things private

From a **security** POV : Potential …

... breaches!

Reflection Access Control

But not too much

Too much bookkeeping…

No bookkeeping!

Should be transparent to developers

Retain Reflection Power

Reflection-Proof Access Control

Reflection breaks walls and rules

Access control of reflective operations
-> **Transparent** to developers
-> Retain **power** of reflection
-> **Reflection-proof**

Conflation

The base-level and the meta-level are **mixed**

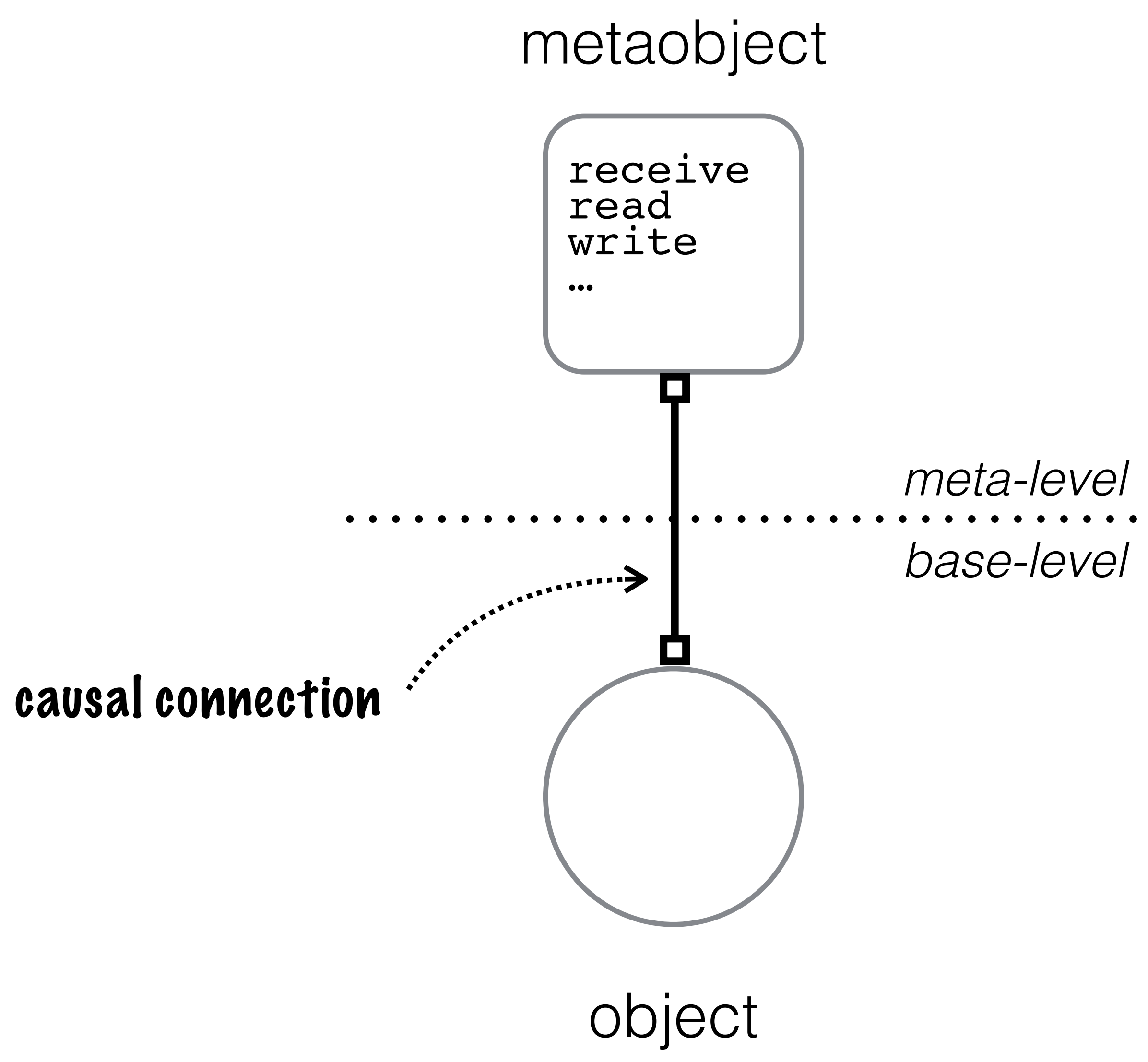
```
>> c := OrderedCollection new.
an OrderedCollection()
>> c add: 10. c
an OrderedCollection(10)
>> c instVarNamed: #array
#(10 nil nil nil nil nil nil nil nil nil)
```

Stratification

The base-level and the meta-level are **separated**

```
>> c := OrderedCollection new.
an OrderedCollection()
>> c add: 10. c
an OrderedCollection(10)
>> c meta instVarNamed: #array
#(10 nil nil nil nil nil nil nil nil nil)
```

Metaobject Protocol



In a MOP:

Control access to **reflective operations**

$$\downarrow$$

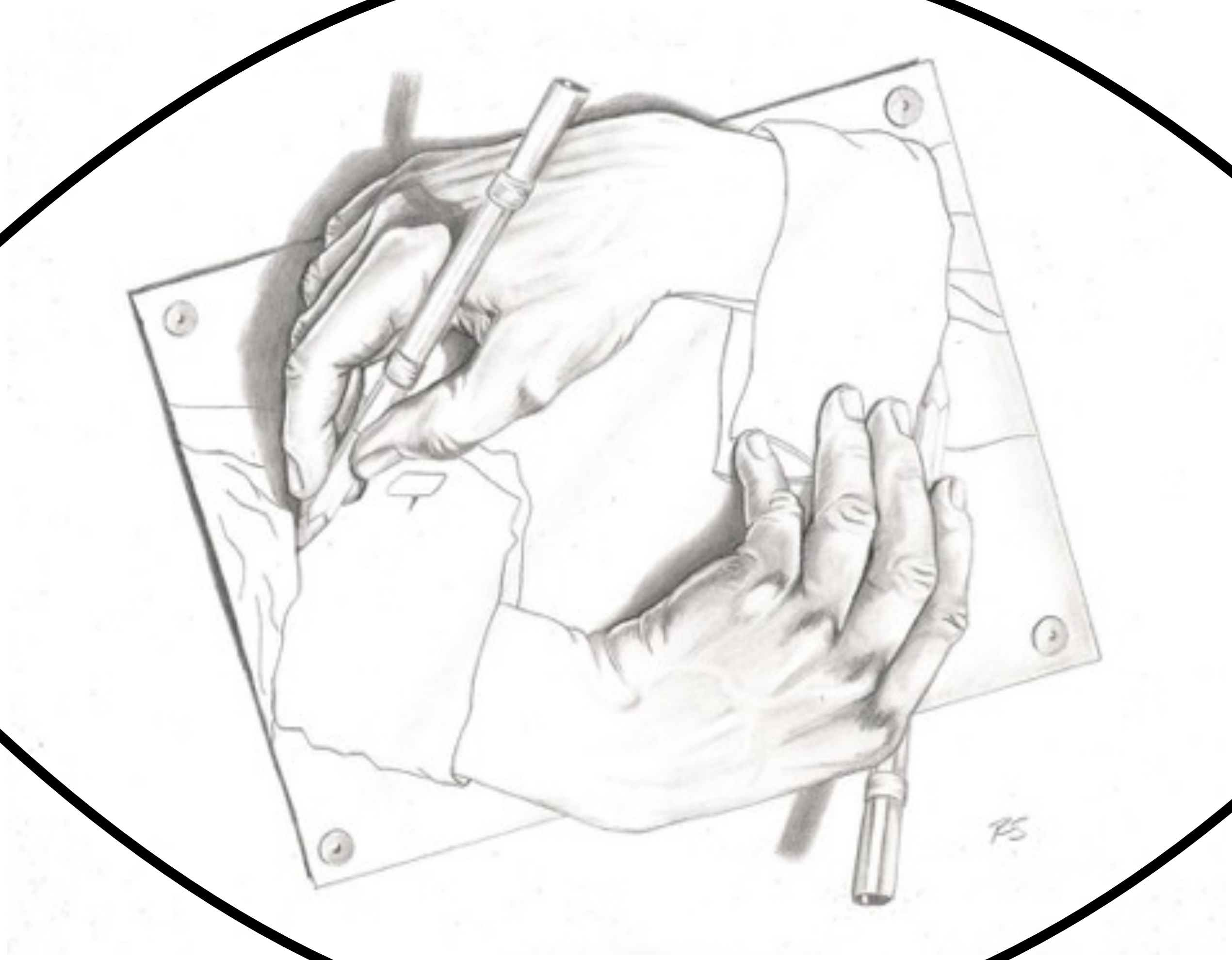
Control access to **metaobjects**

Idea

Reflection can implement security mechanisms!

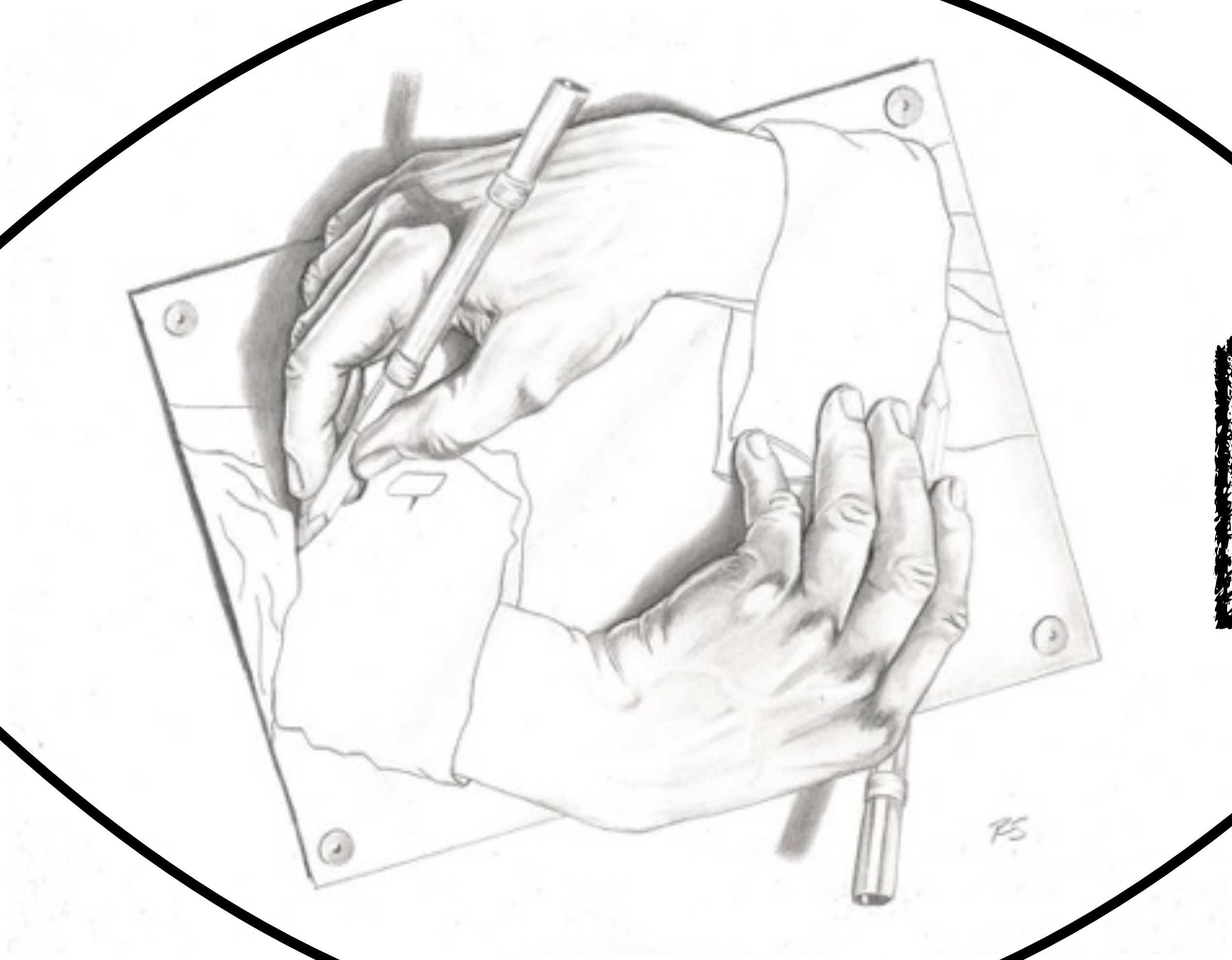
Metacircular Security

$$\begin{tikzcd}
\text{Reflection} \arrow[r, bend left, "implements"] & \text{Security mechanism(s)} \arrow[l, bend left, "secures"]
\end{tikzcd}$$



Metacircular Security

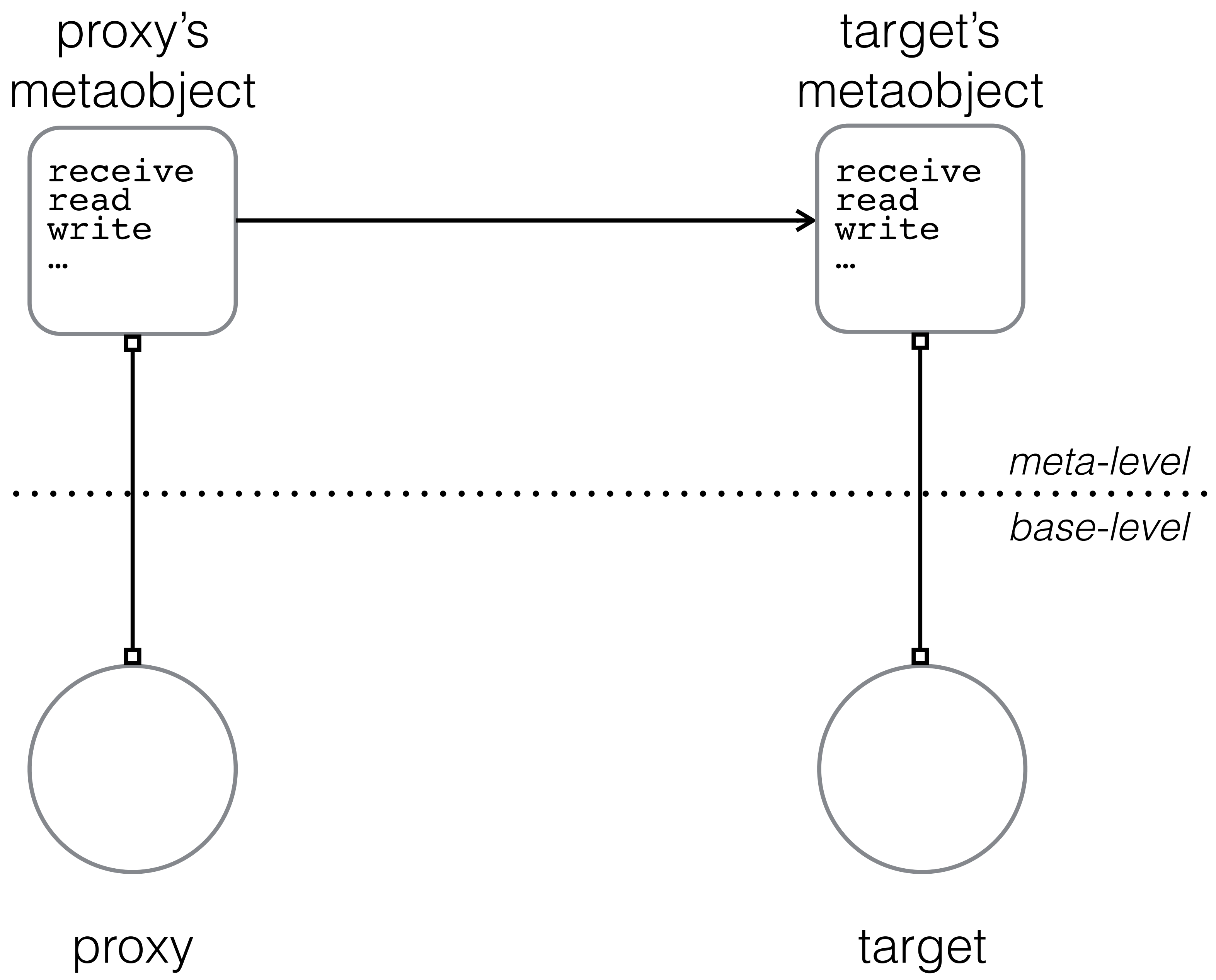
$$\begin{tikzcd}
\text{Reflection} \arrow[r, bend left, "\textit{implements}"] & \boxed{\textbf{Security mechanism(s)}} \arrow[l, bend left, "\textit{secures}"]
\end{tikzcd}$$



Proxies for Access Control

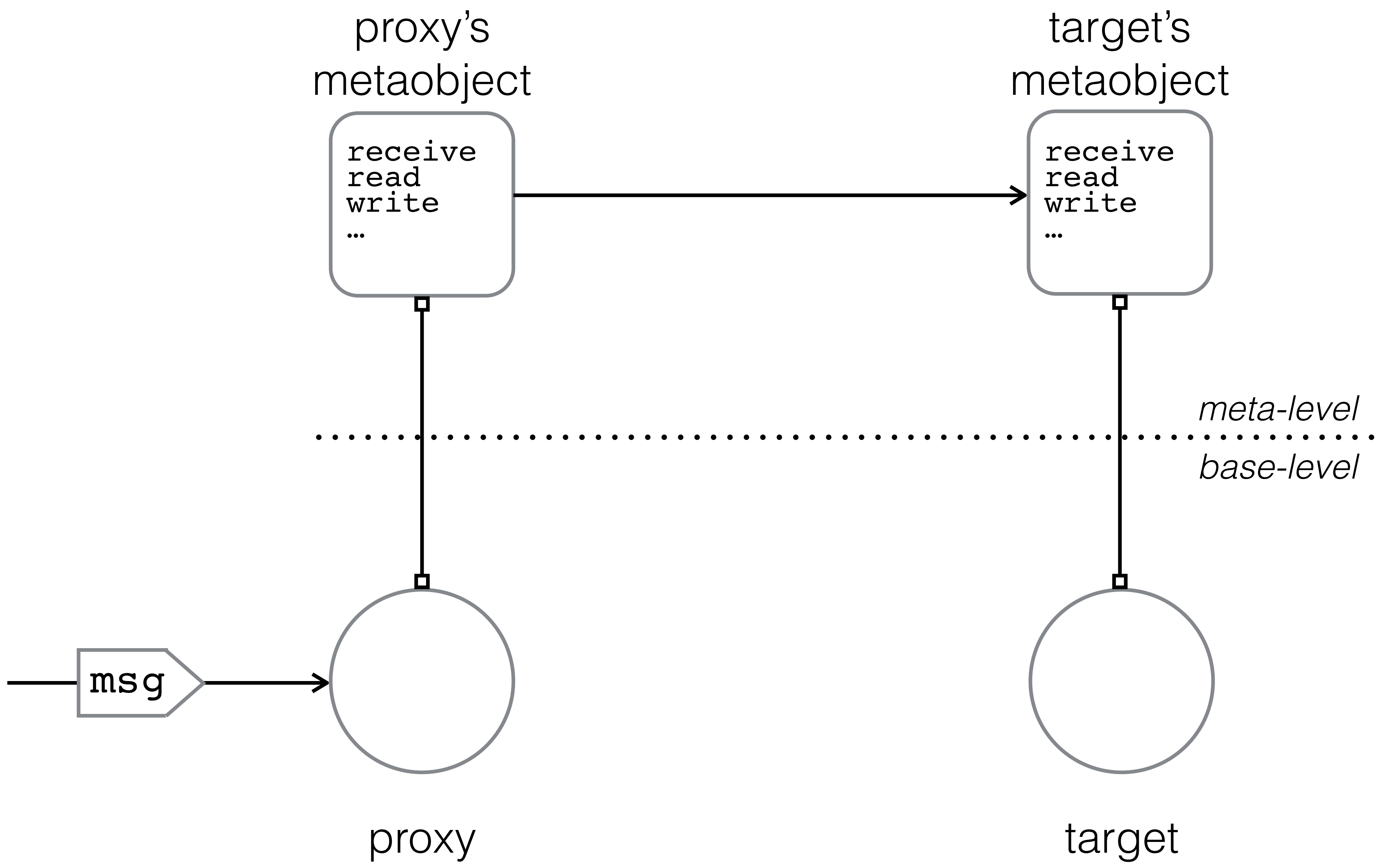
Proxies

Fine-grained intercession



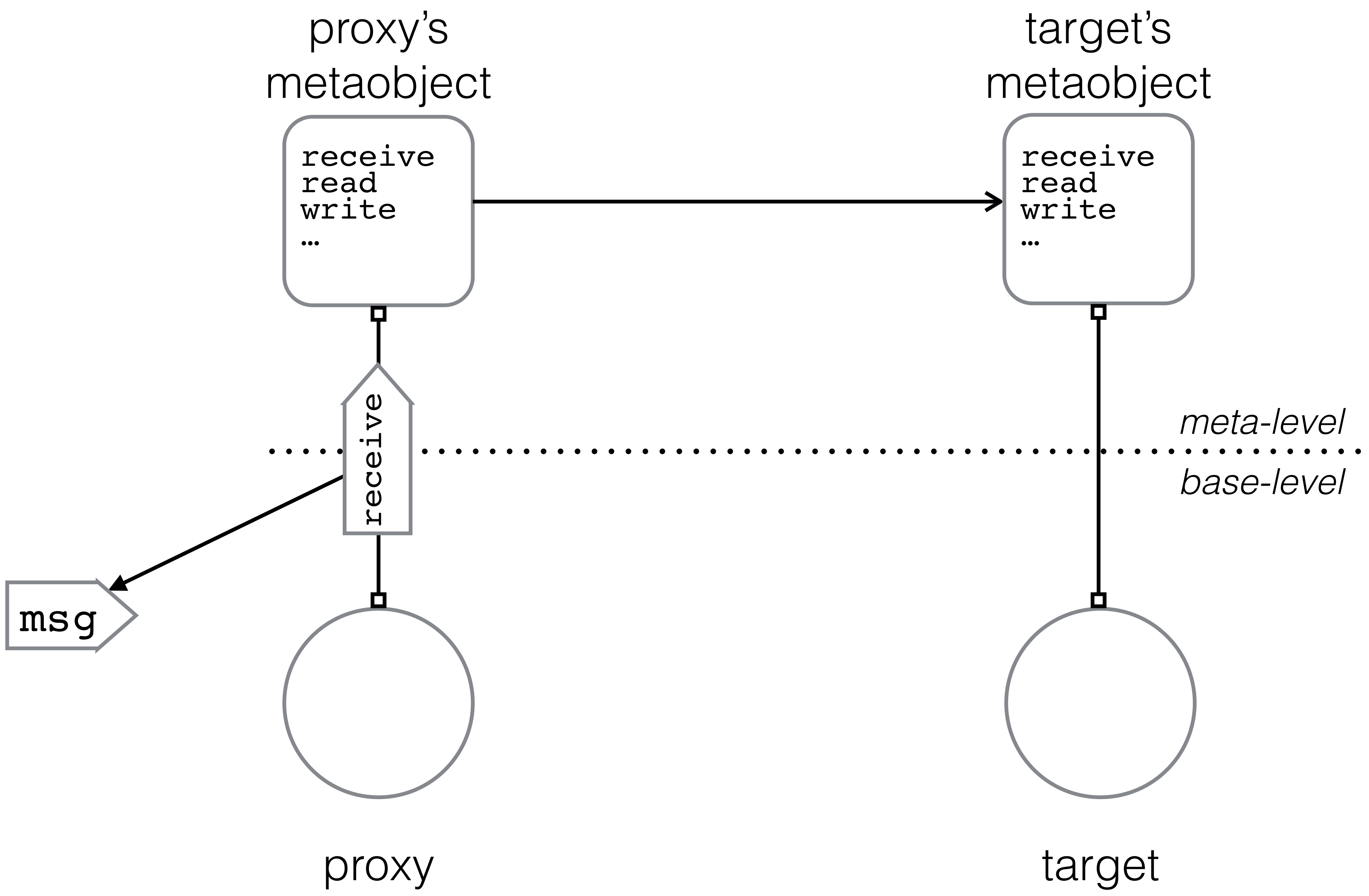
Proxies

Fine-grained intercession



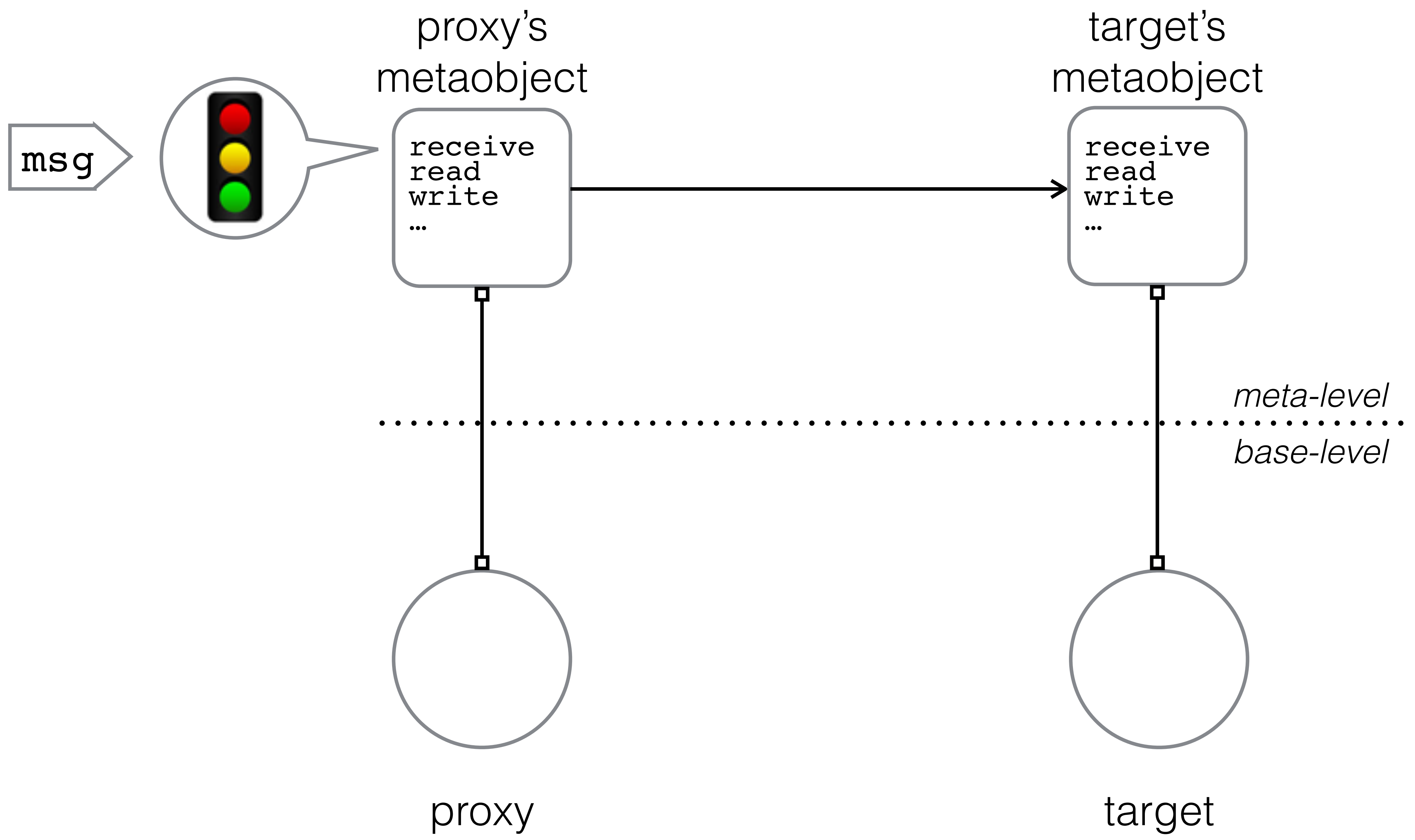
Proxies

Fine-grained intercession



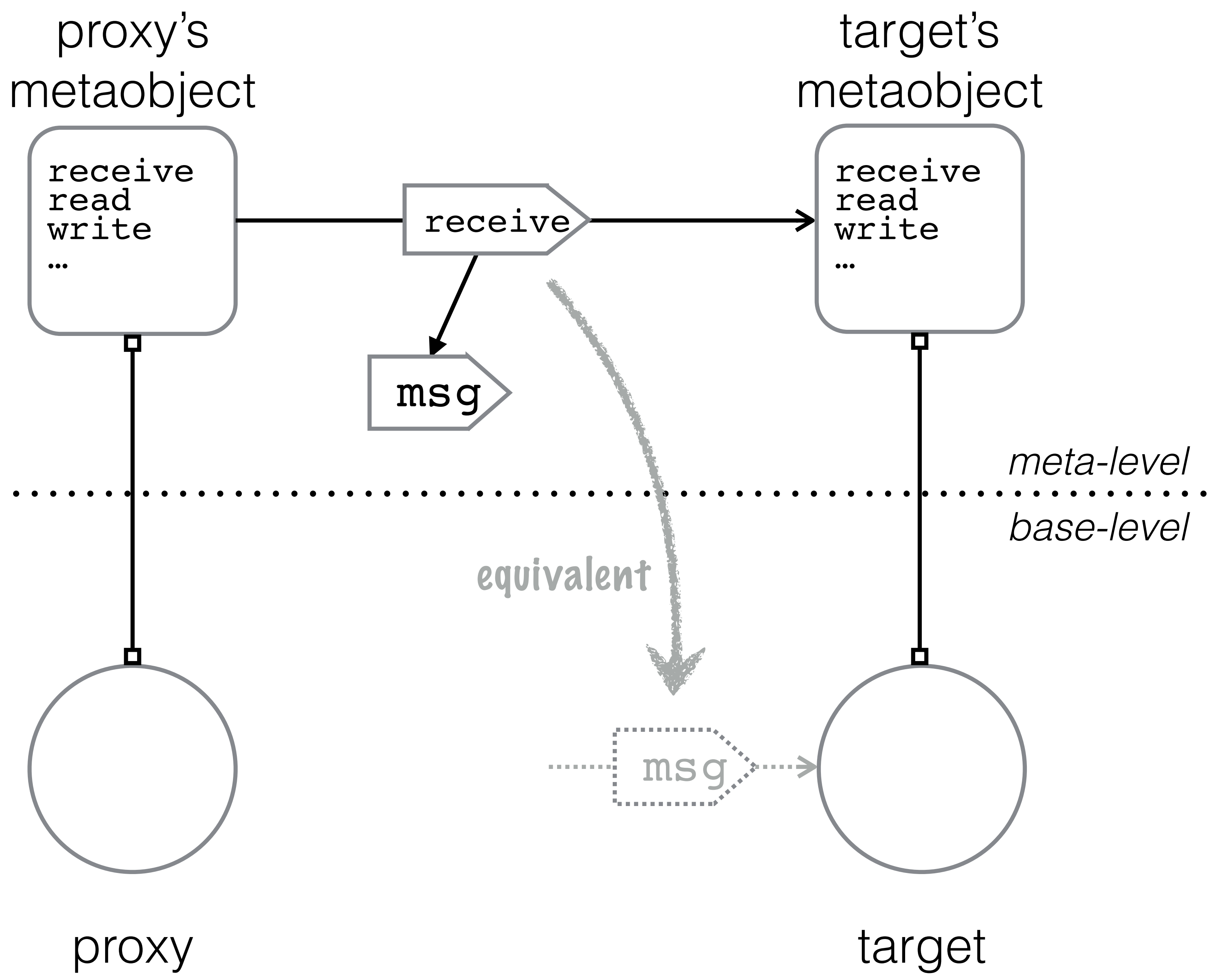
Proxies

Fine-grained intercession



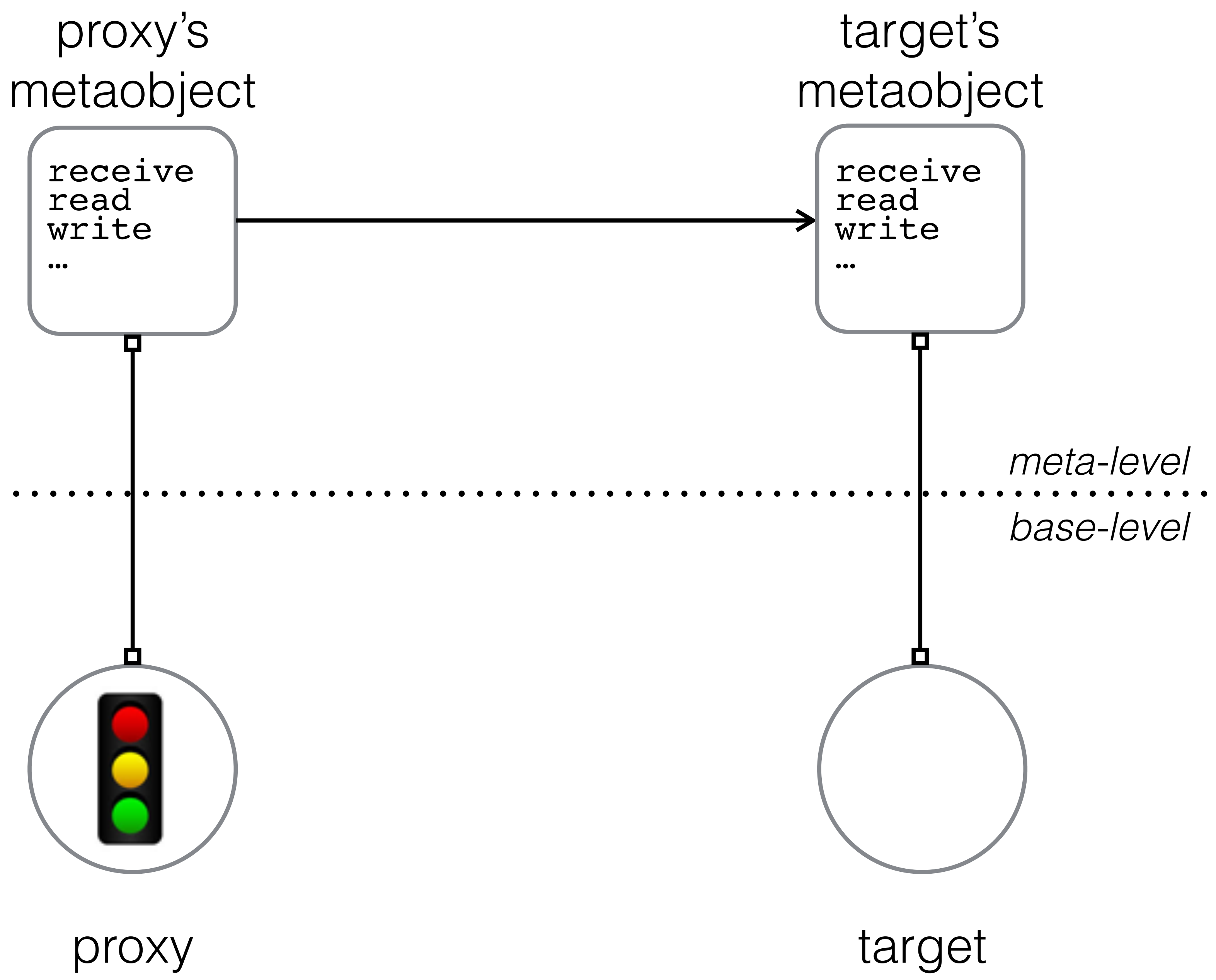
Proxies

Fine-grained intercession



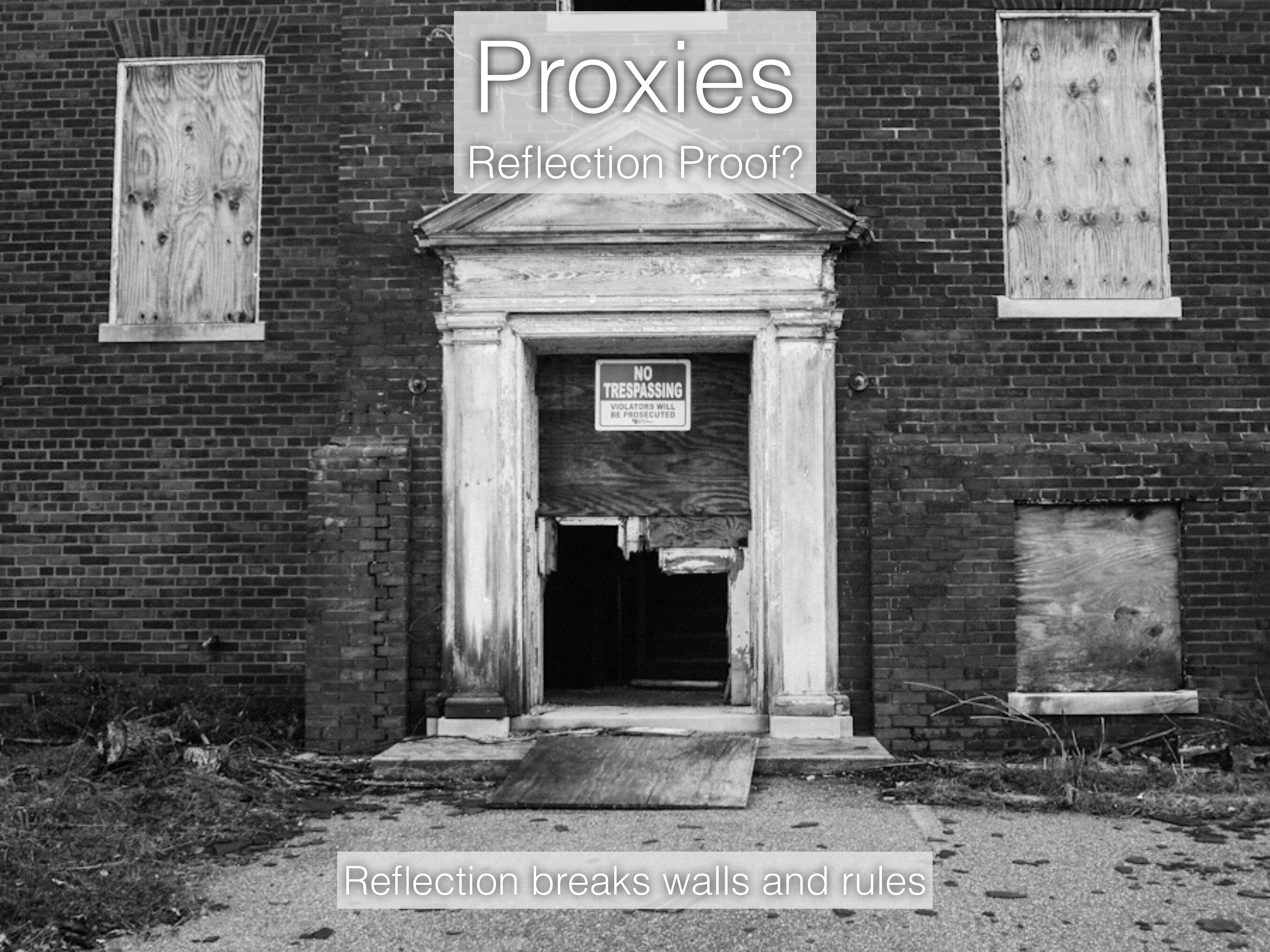
Proxies

Access Control Mechanism



Proxies

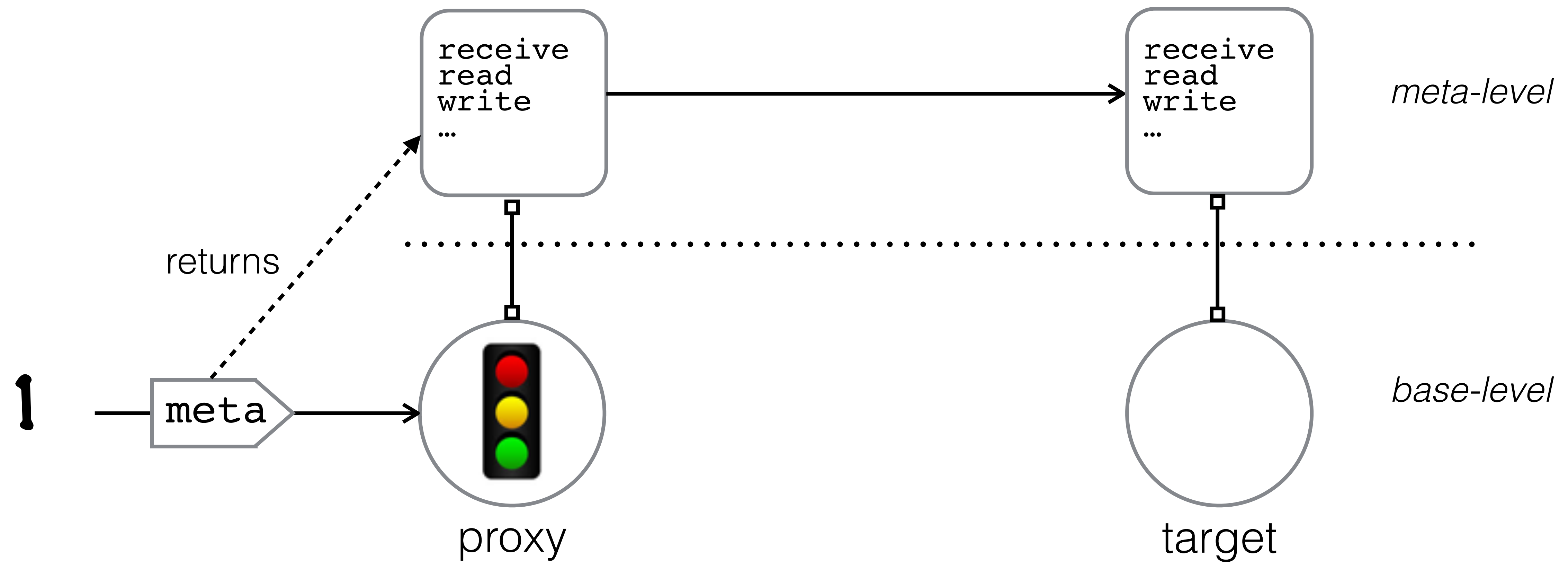
Reflection Proof?



Reflection breaks walls and rules

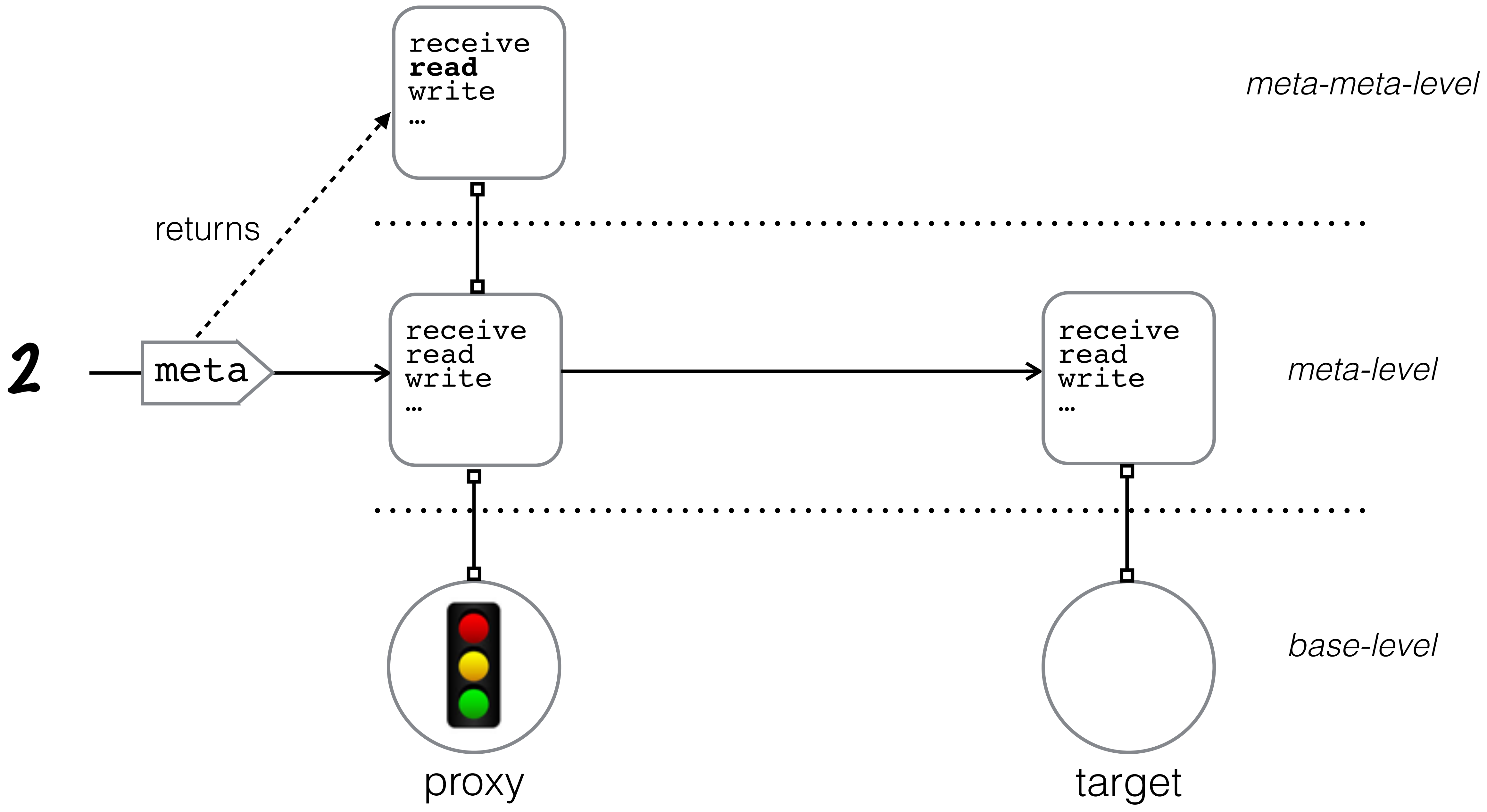
Proxies

Reflection Proof?



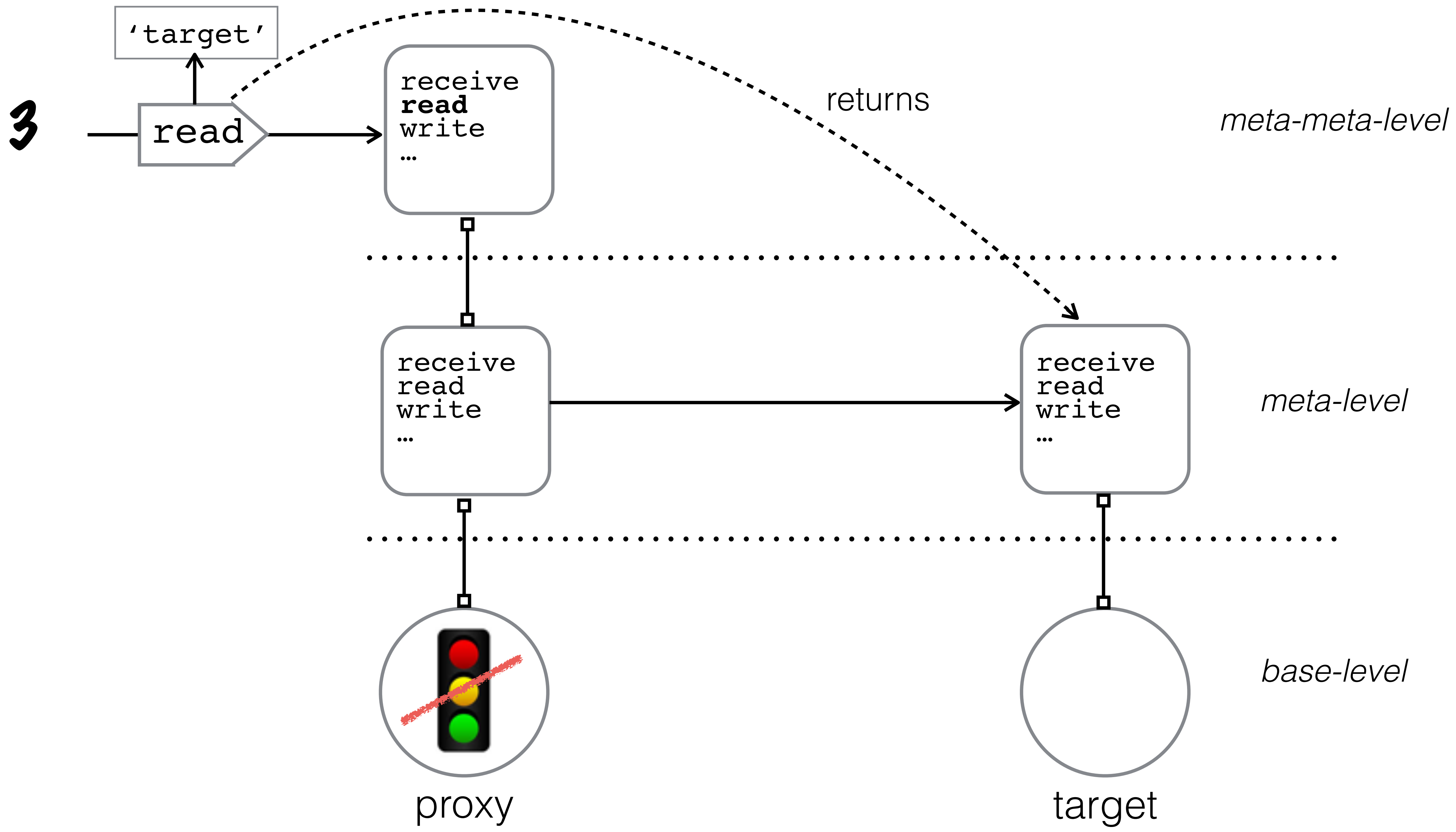
Proxies

Reflection Proof?

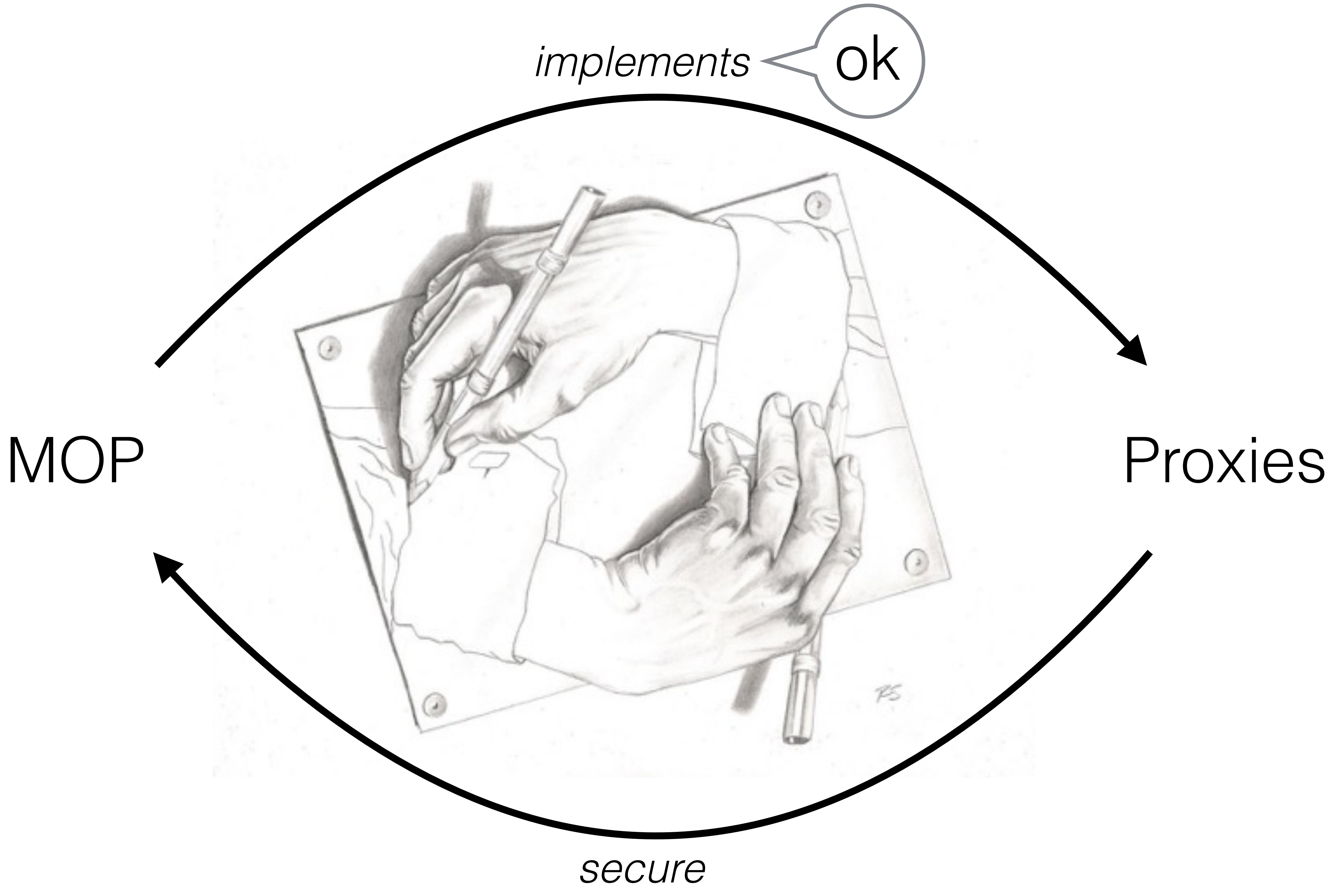


Proxies

Reflection Proof?



Metacircular Security

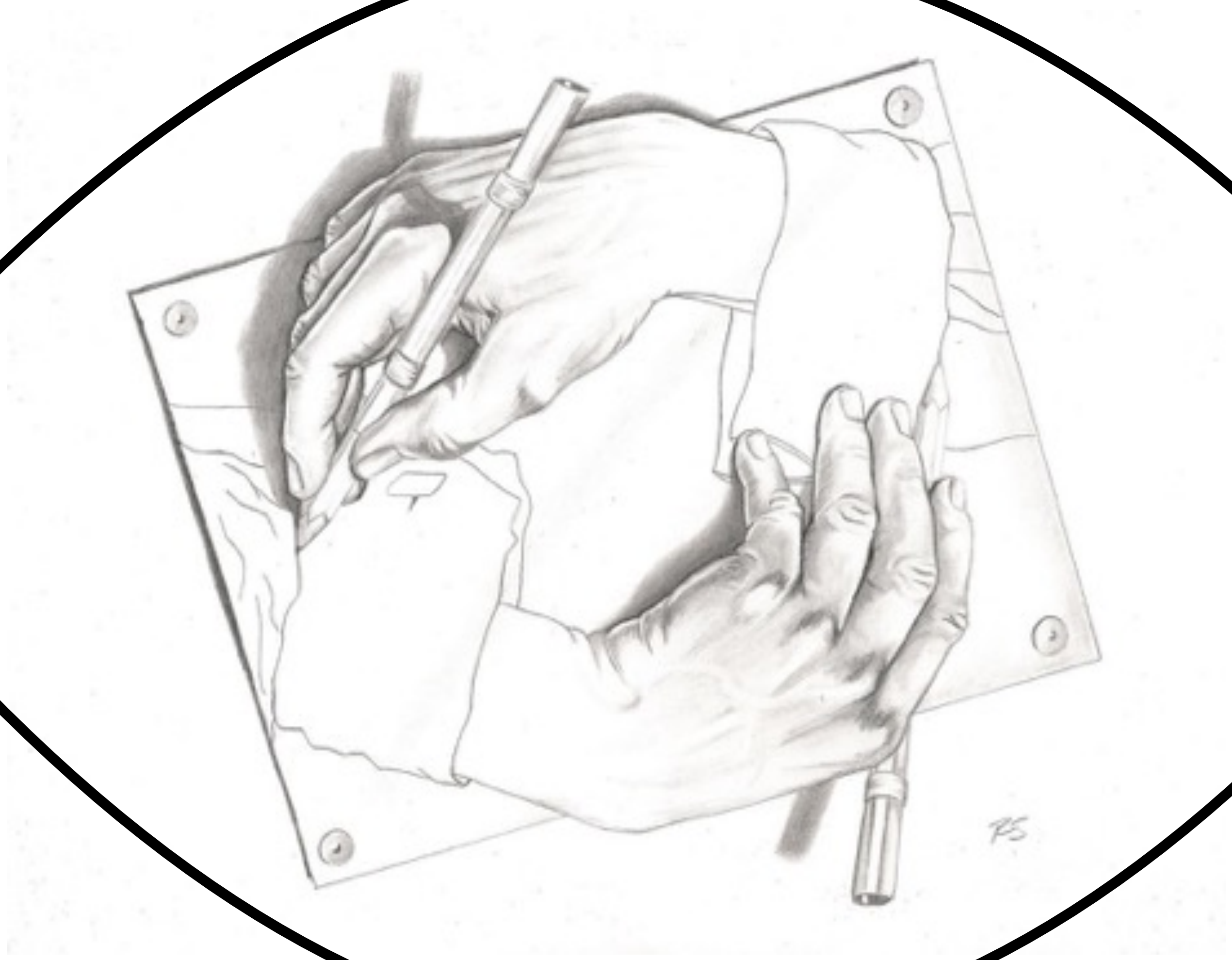


Metacircular Security

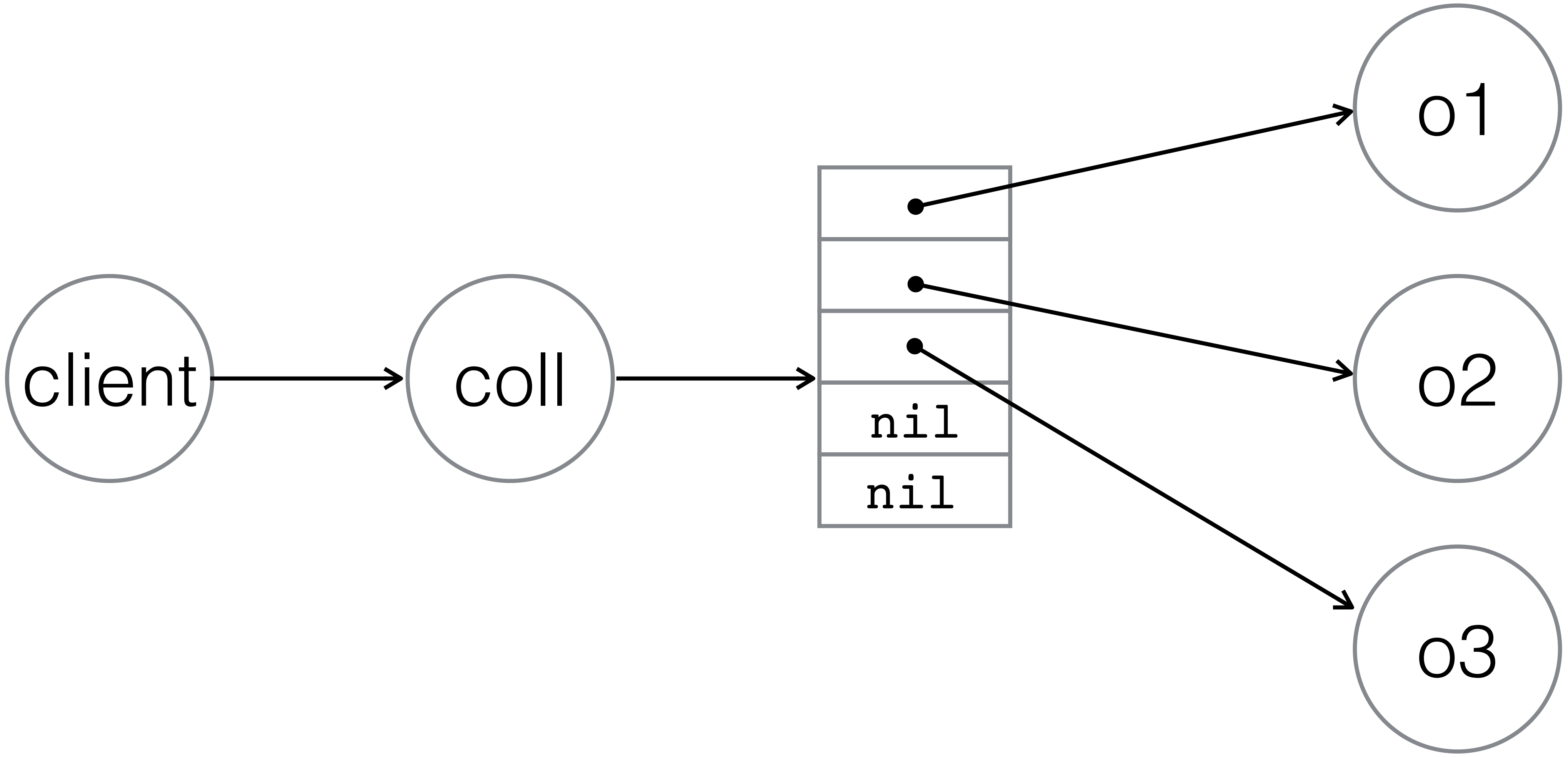
$$\begin{tikzcd}
\text{MOP} \arrow[r, bend left, "\textit{implements}"] & \text{Proxies} \arrow[l, bend left, "\textit{secure}"]
\end{tikzcd}$$

ok

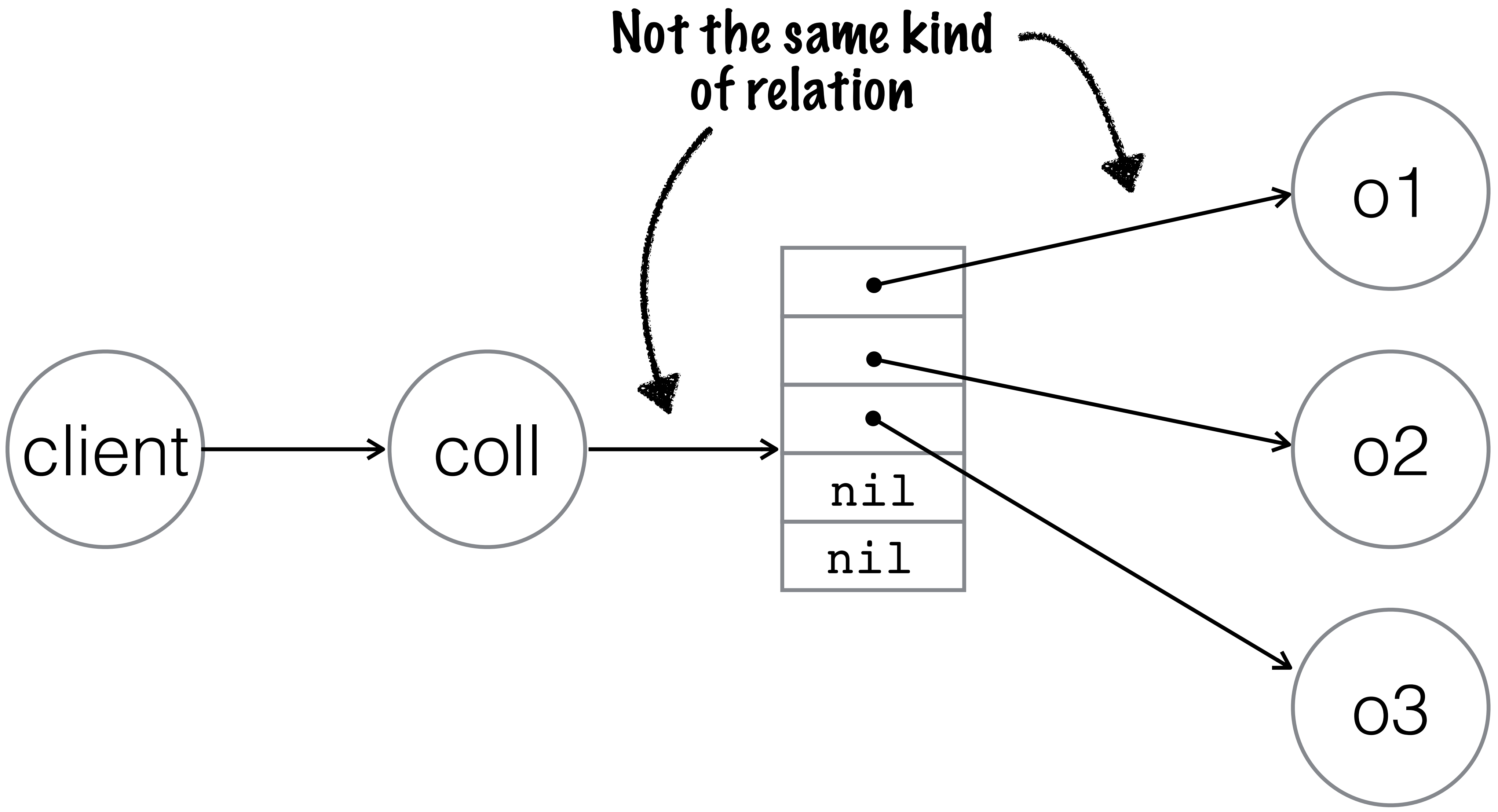
What policy?



OrderedCollection Revisited



OrderedCollection Revisited

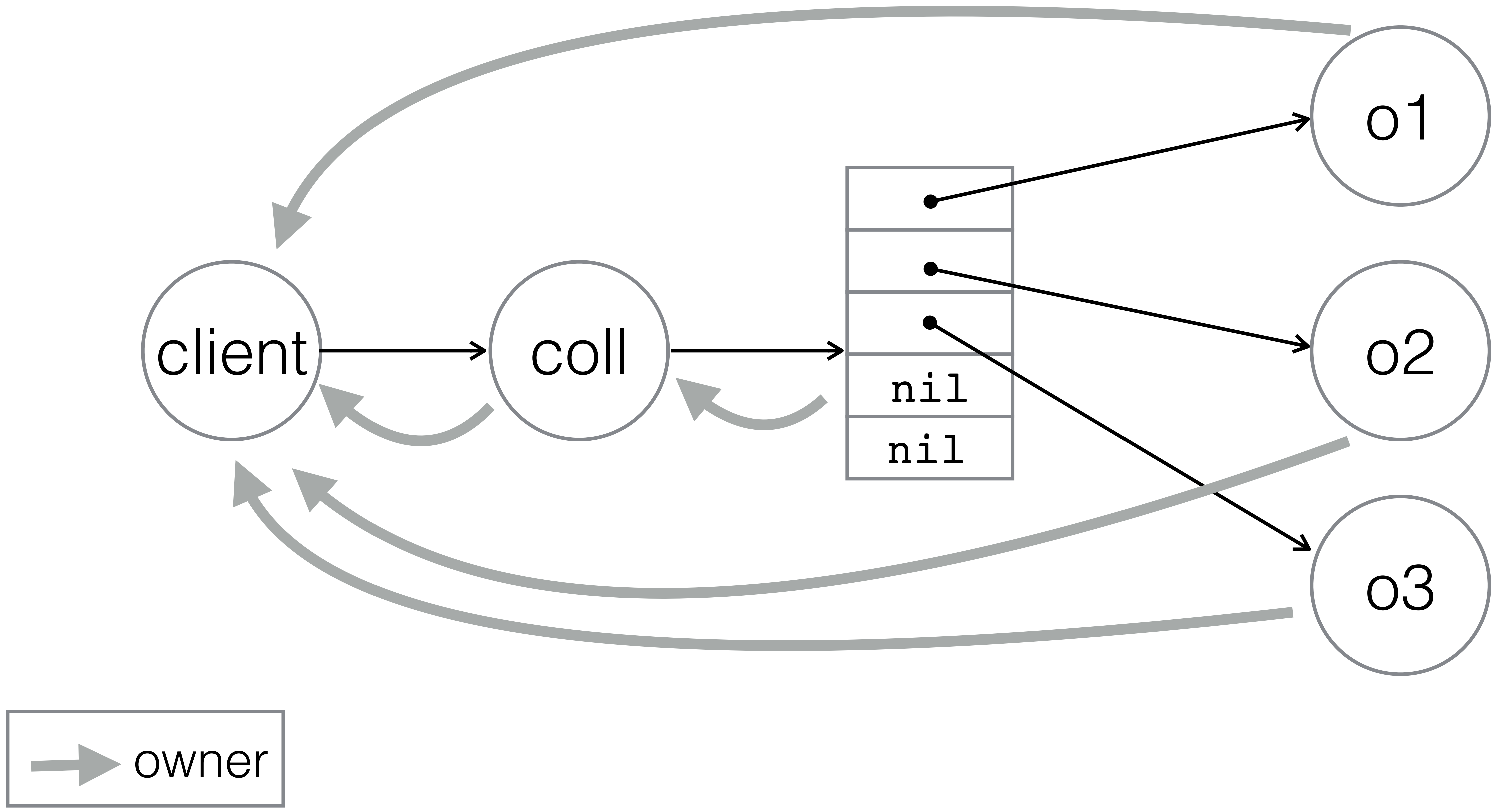


Object Ownership:
Access Control Policy to Metaobject

- An object owns its metaobject
- The ownership relation is transitive
- A client object can access the metaobjects of the objects it owns
- Other objects have a limited access
  —> implemented by a proxy

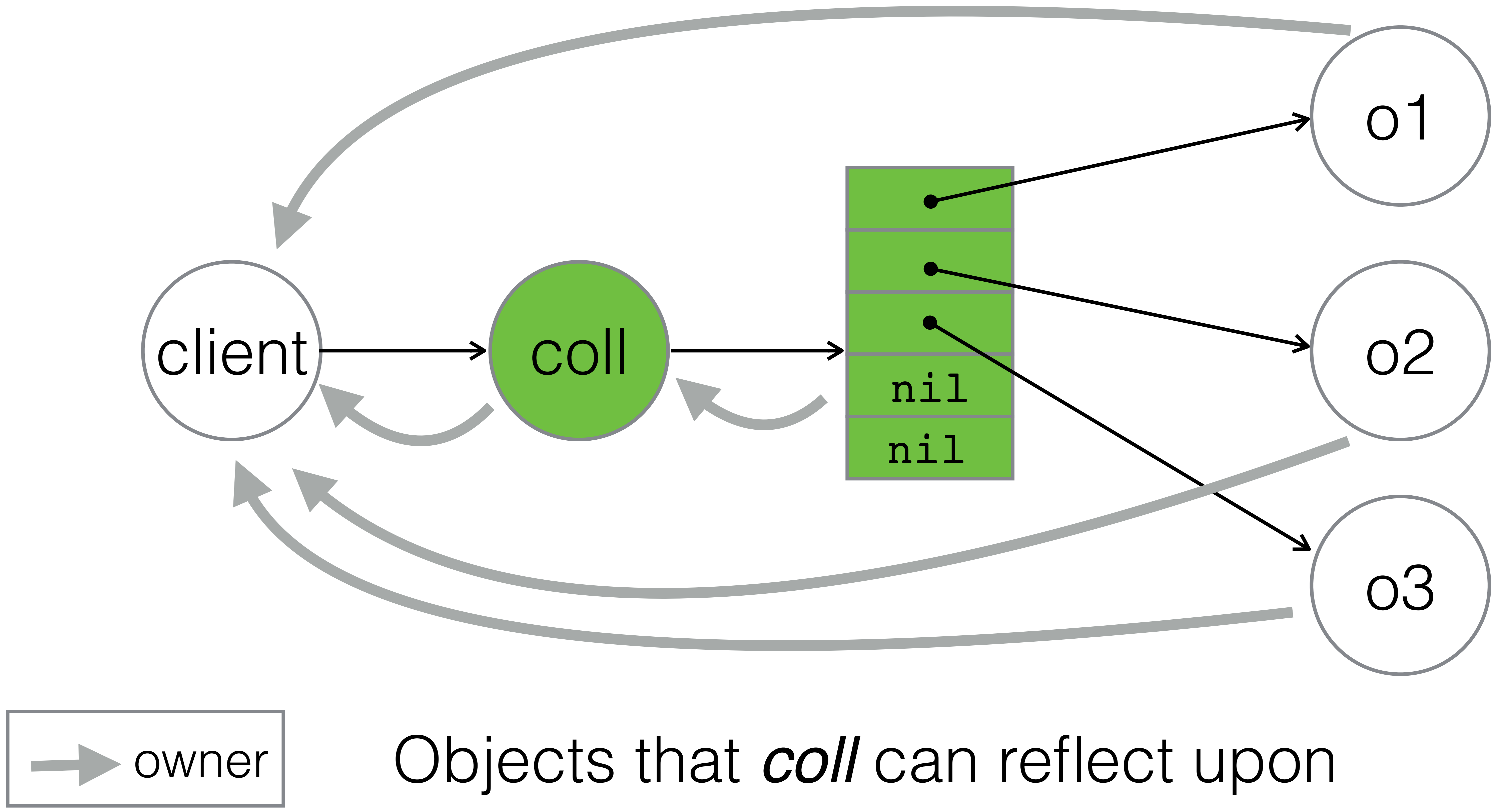
Object Ownership:

Access Control Policy to Metaobject



Object Ownership:

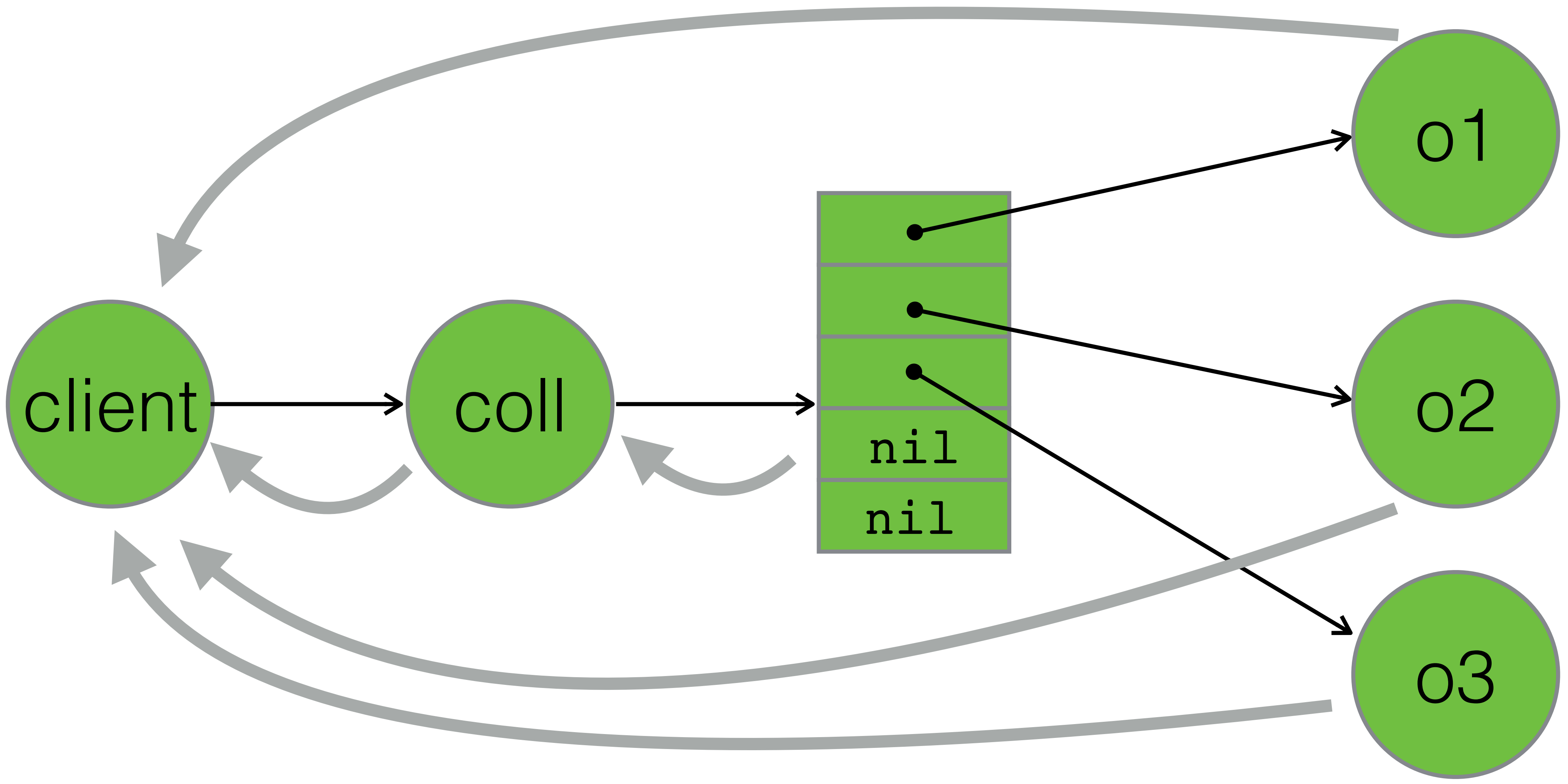
Access Control Policy to Metaobject



Objects that *coll* can reflect upon

Object Ownership:

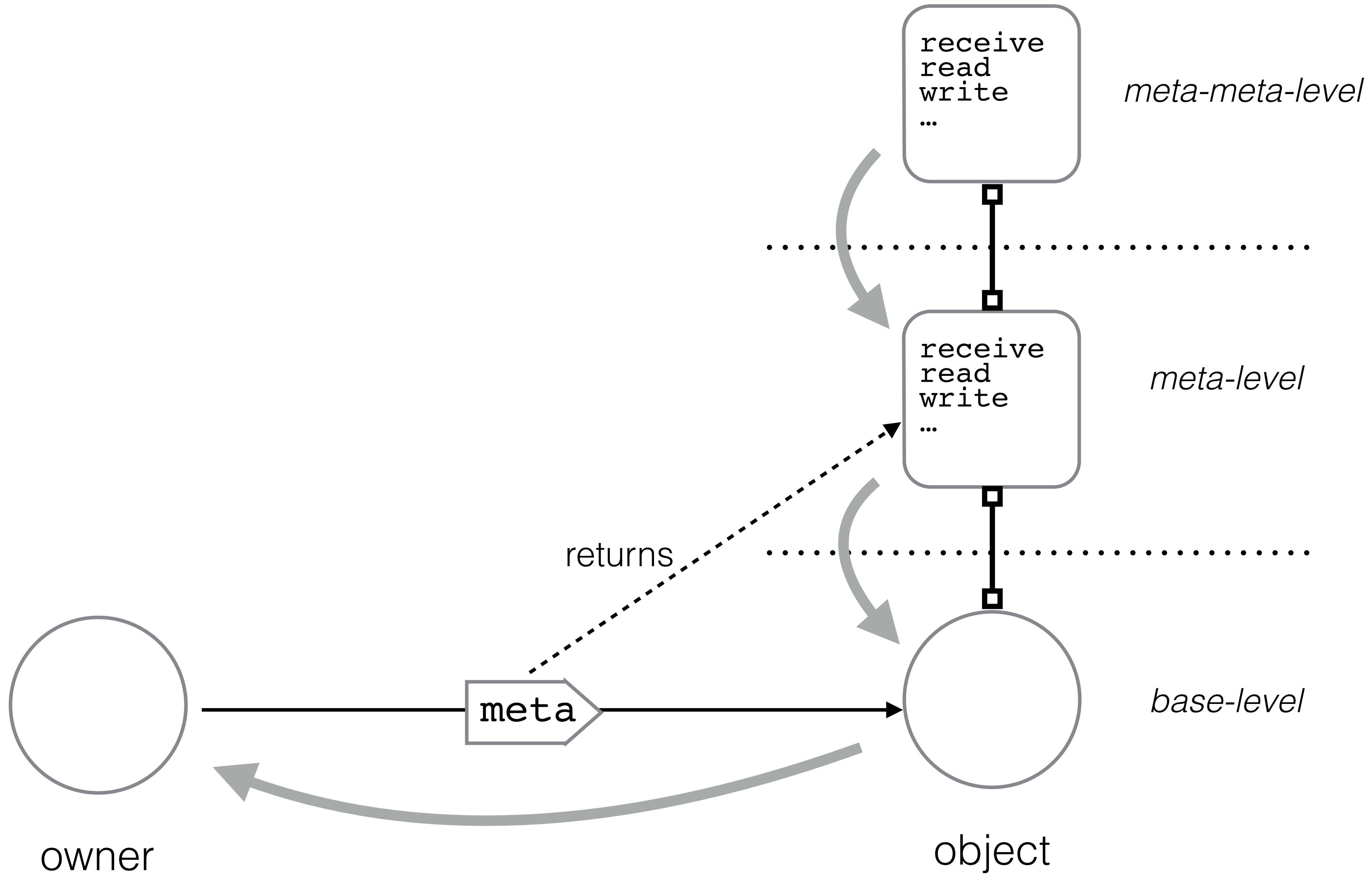
Access Control Policy to Metaobject



Objects that ***client*** can reflect upon

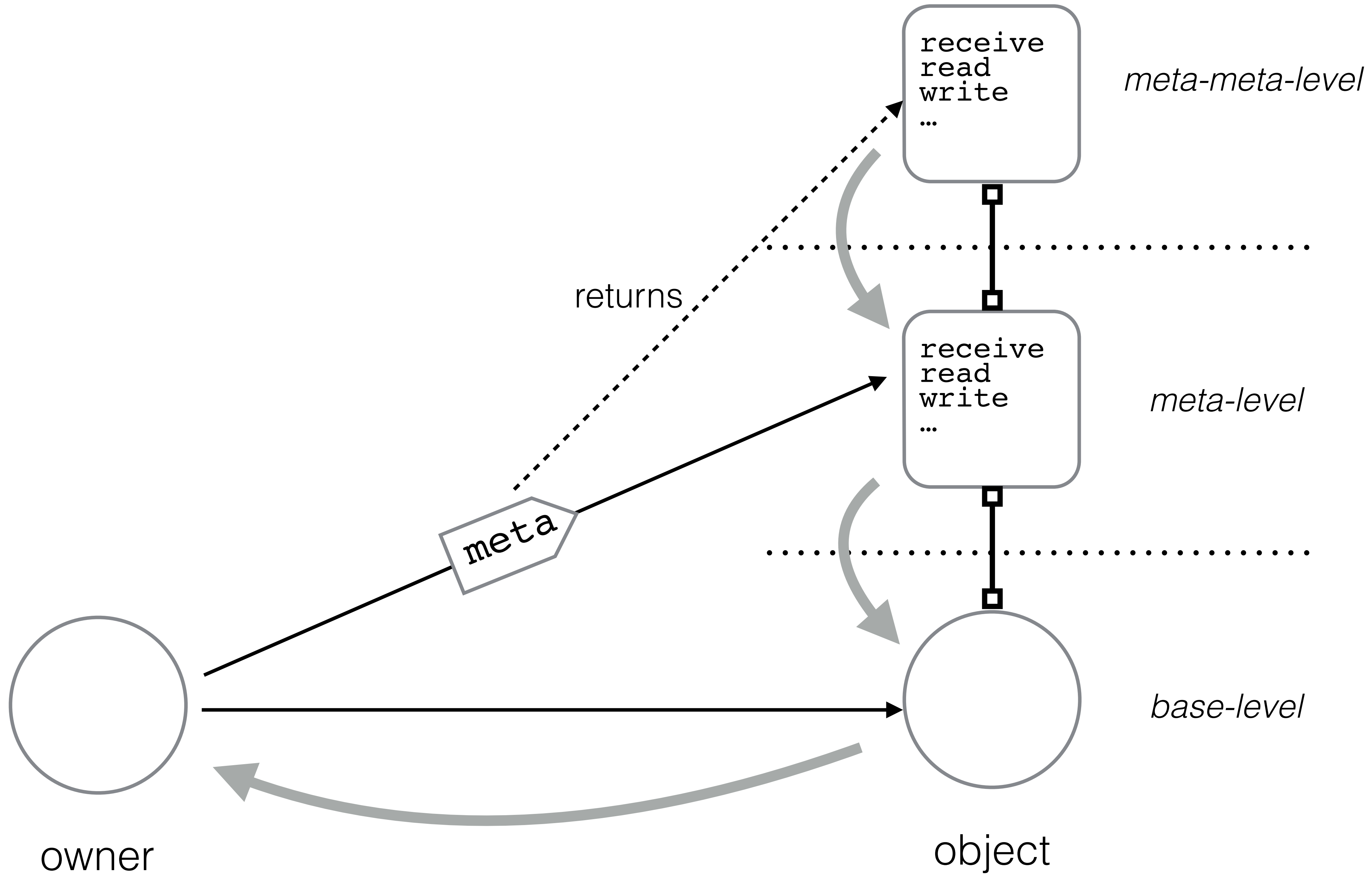
Object Ownership:

Access Control Policy to Metaobject



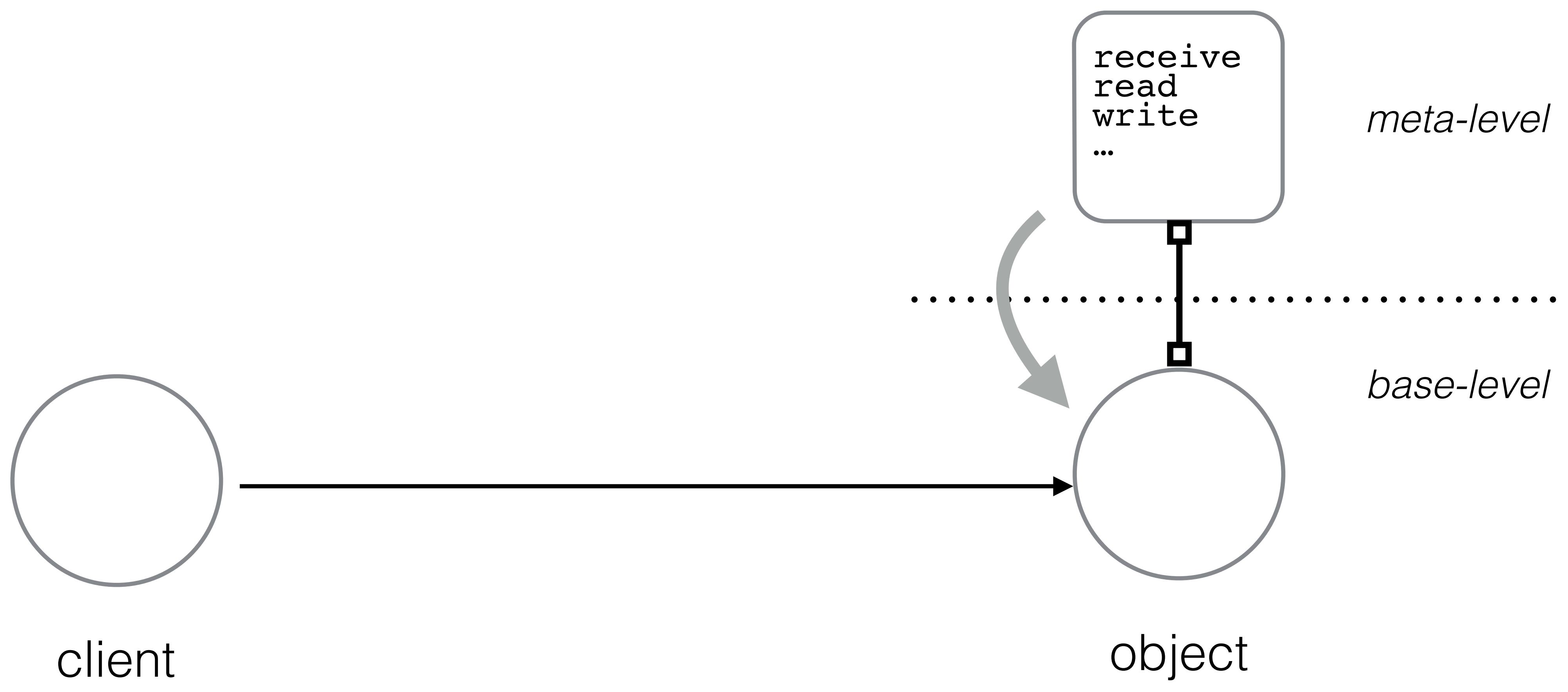
Object Ownership:

Access Control Policy to Metaobject



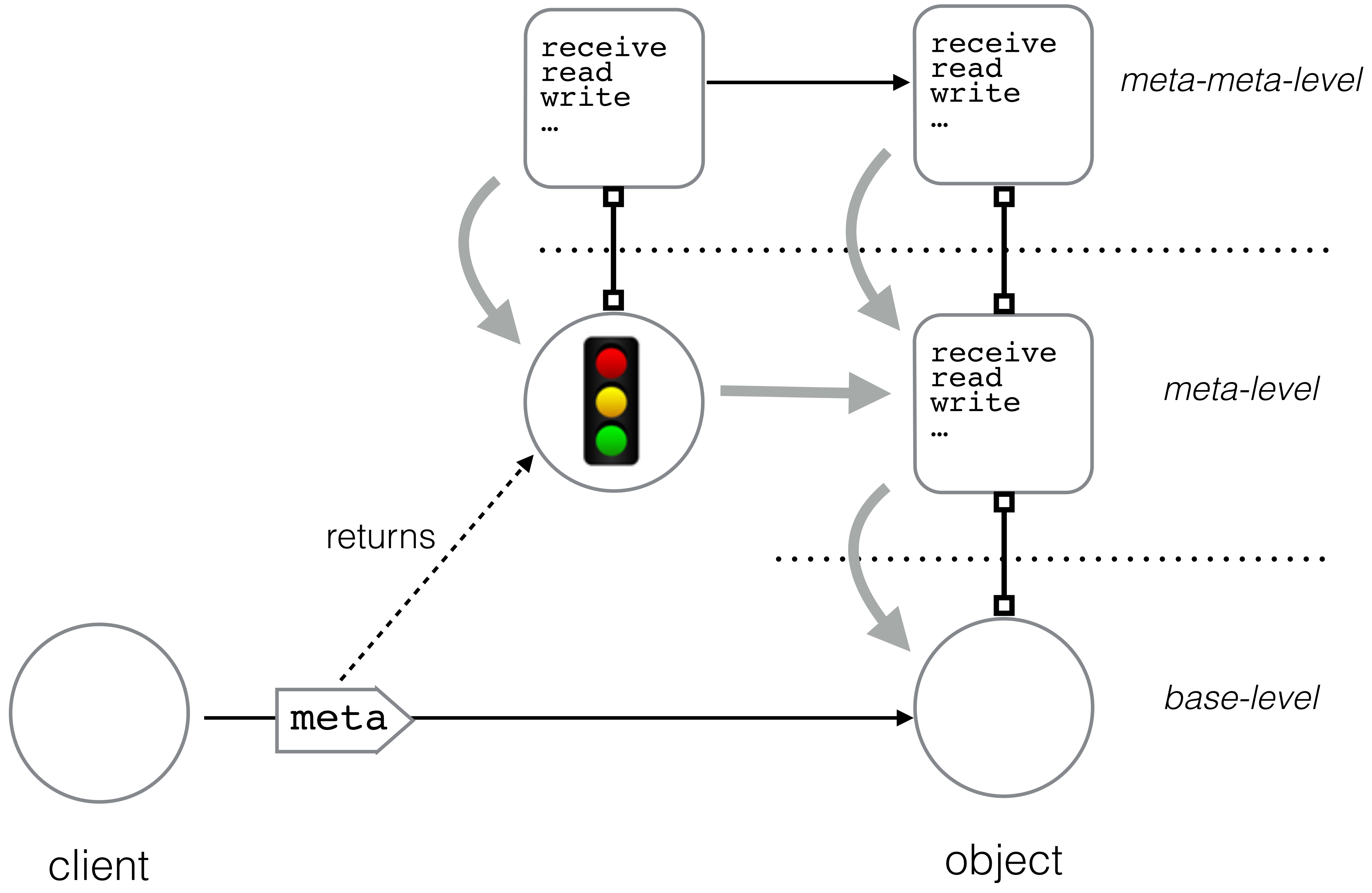
Object Ownership:

Access Control Policy to Metaobject



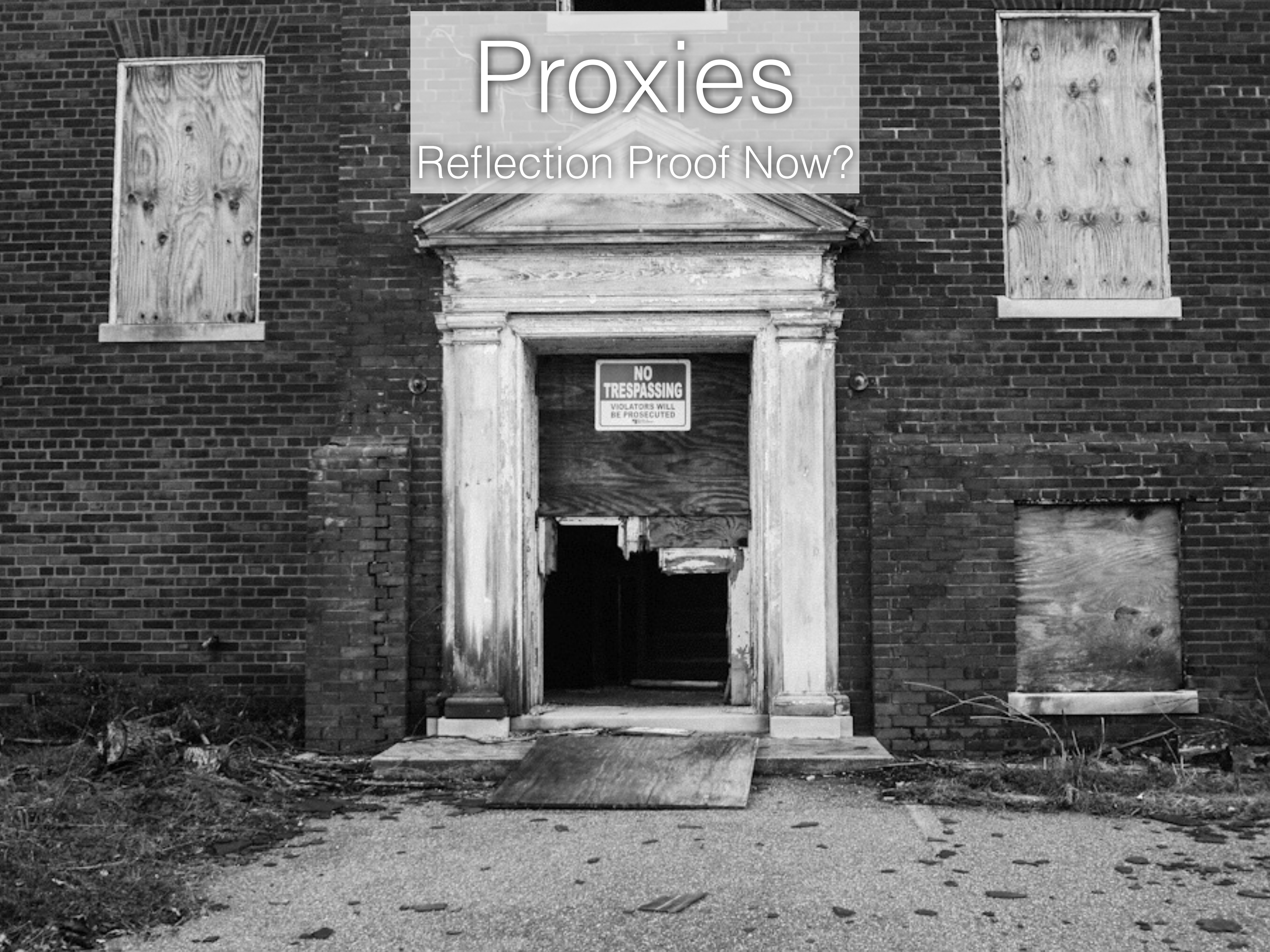
Object Ownership:

Access Control Policy to Metaobject



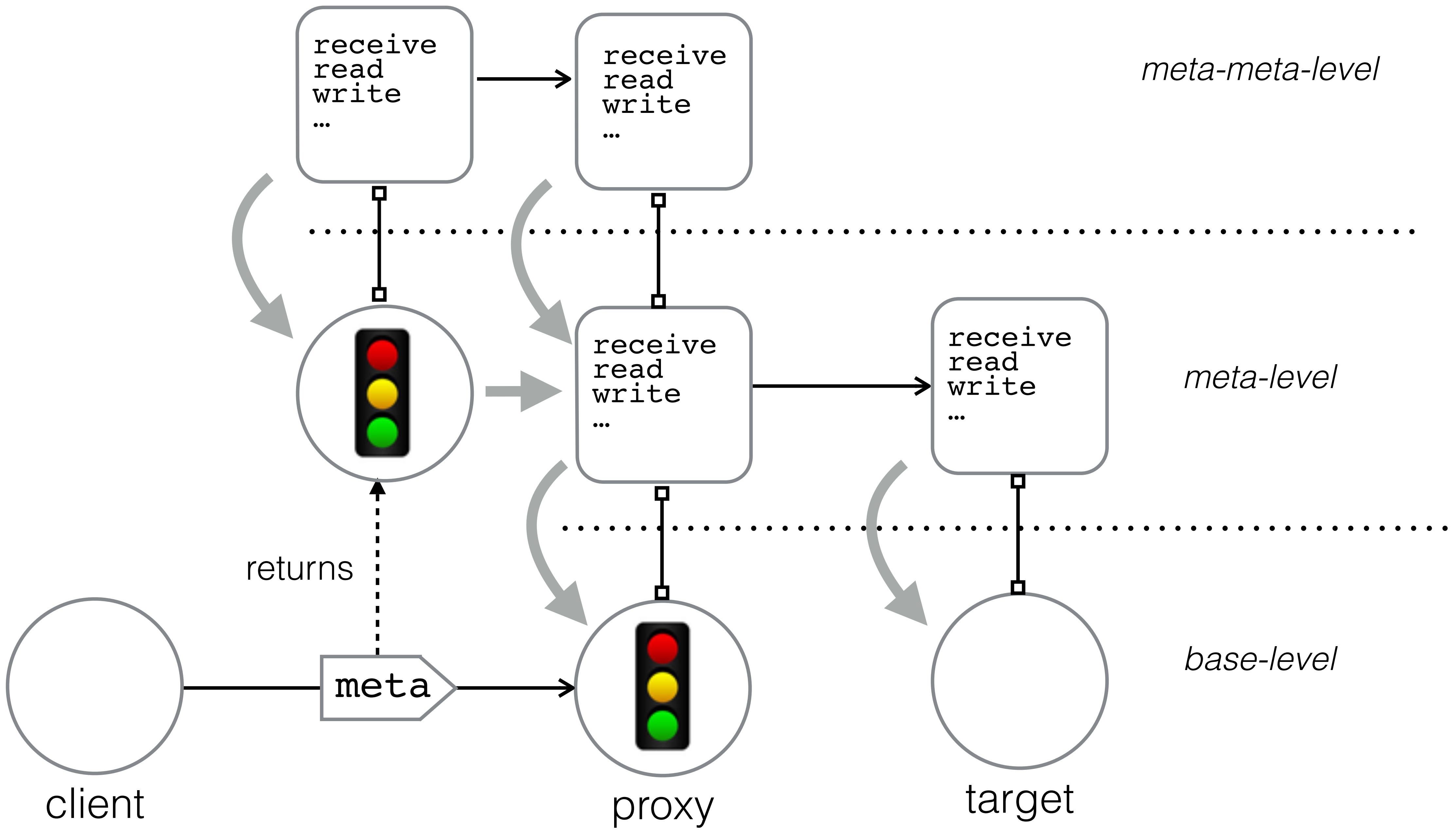
Proxies

Reflection Proof Now?



Proxies

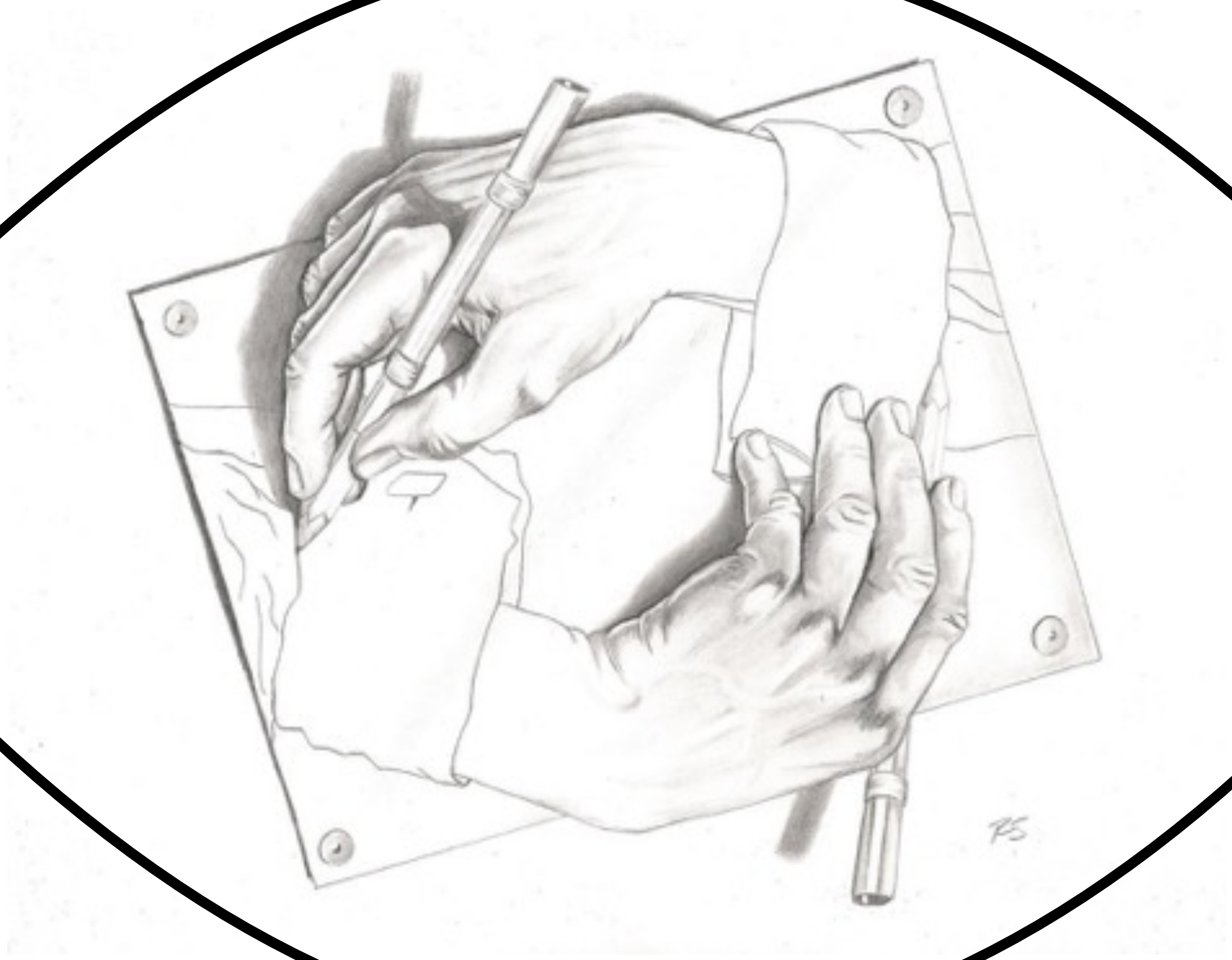
Reflection Proof Now?



Access control of reflective operations
-> **Transparent** to developers
-> Retain (most) **power** of reflection
-> **Reflection-proof**

Metacircular Security

$$\begin{tikzcd}
\text{MOP} \arrow[r, bend left, "\textit{implements}"] & \text{Proxies} \arrow[l, bend left, "\begin{array}{c}\textit{secure}\\ \textit{with an ownership-based policy}\end{array}"]
\end{tikzcd}$$



Conclusion

Problem

Tension between reflection and security

Solution

	Reflection	—>A MOP
+	A security mechanism	—> Proxies
+	An access control policy	—> Object ownership
=	**Metacircular Security**	