

Mathematical Object Manager

Juliette Mainka

Equipe [M@inline](#), I3S/CNRS/UNSA

What is the MOM?

Content

1. MOM architecture

(Functionality, Components)

2. Use cases

(Handling of promises, Execution of a plan,
Benchmark Sharing)

3. Details on the Implementation

(Technologies, Problem handling, Security issues)

MOM architecture

Functionality

Storing and uniquely **identifying** mathematical objects on the network and **exchanging** such references between nodes.

Why using it?

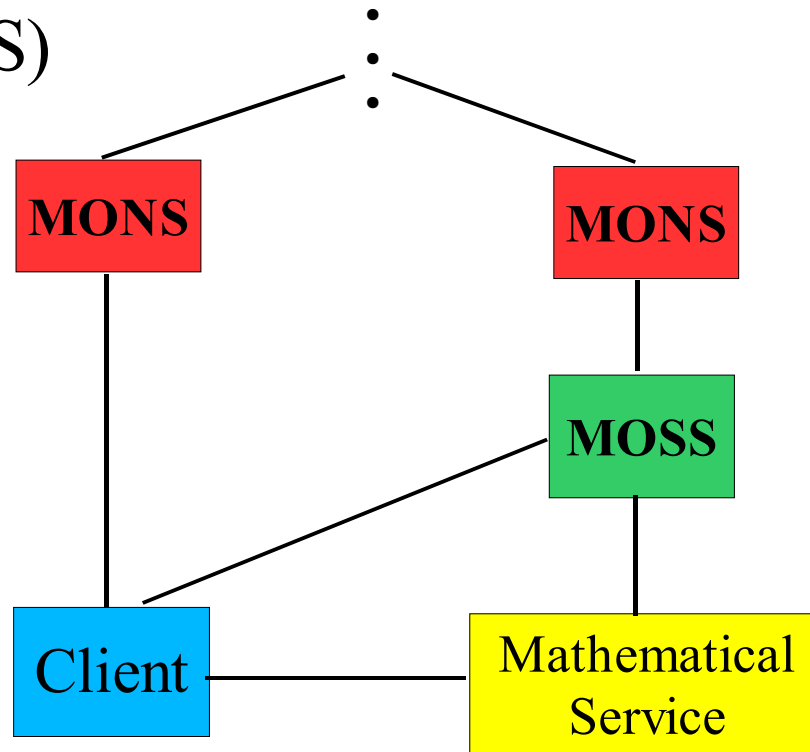
Facilitates work with and handling of mathematical objects:

- persistence and remote accessibility (allows sharing)
- handling of references (reduces object transport, enables promises)

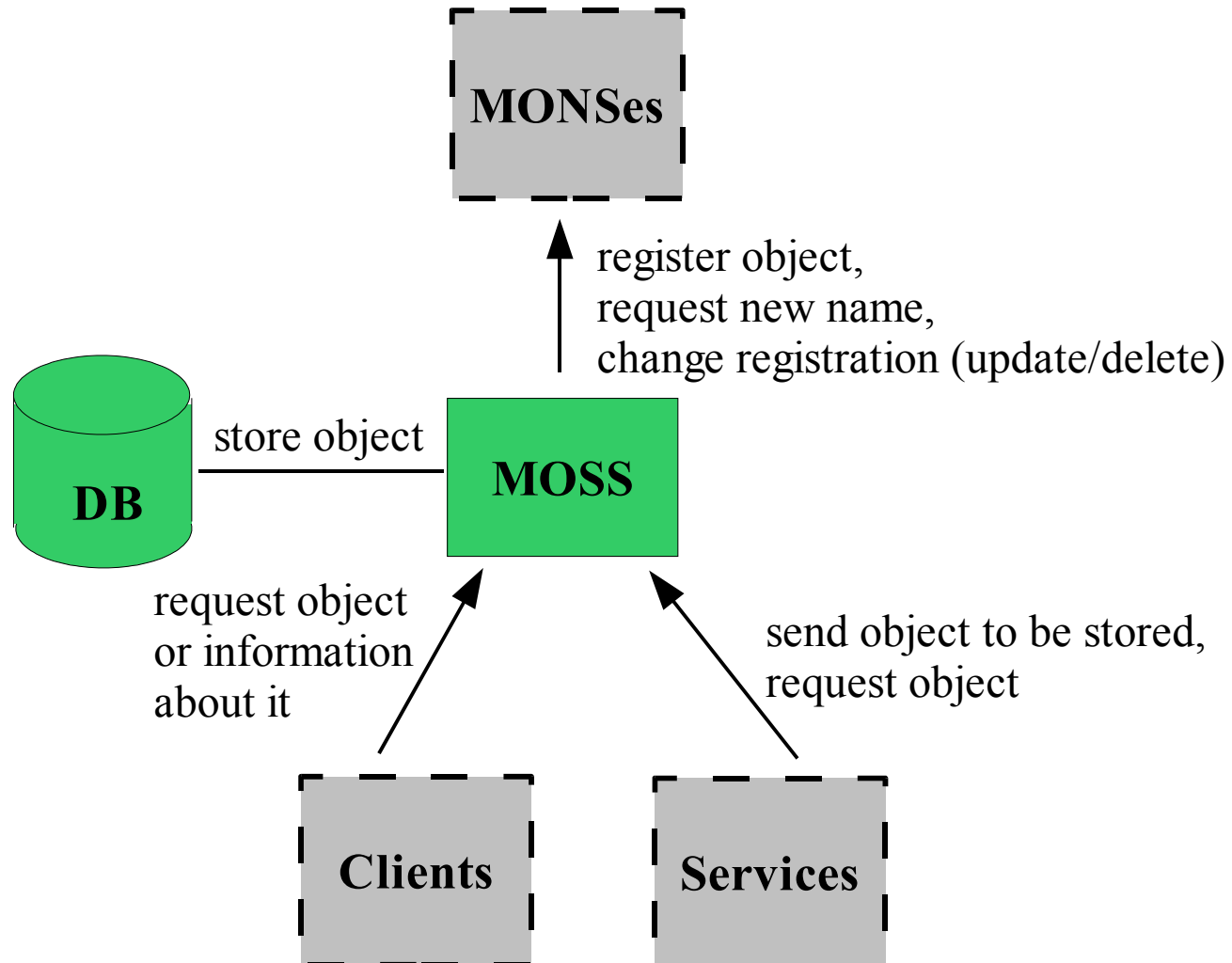
Components of the MOM

Separate MOM functionality into:

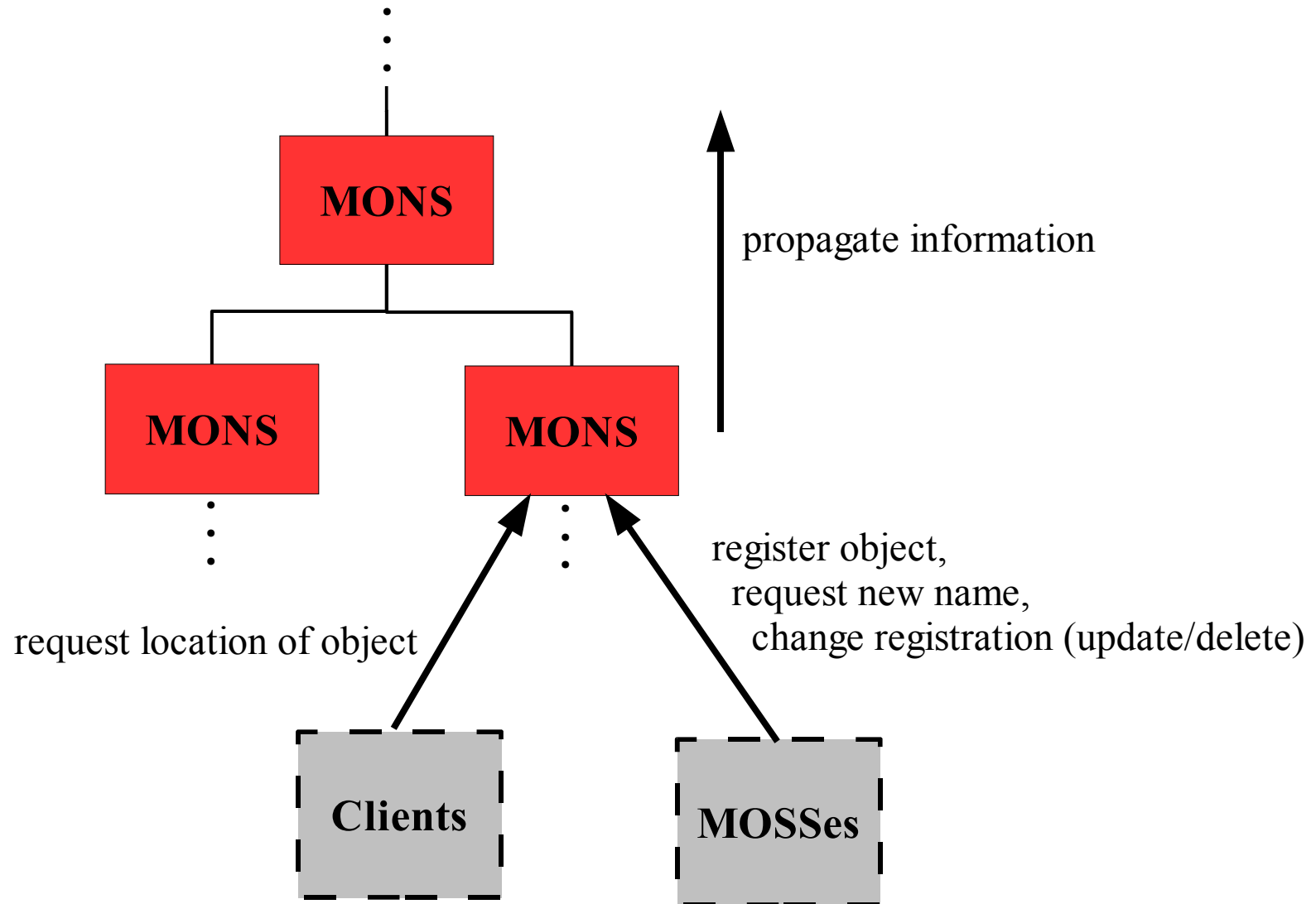
- a Storage Service (MOSS)
- a Naming Service (MONS)



The MOSS

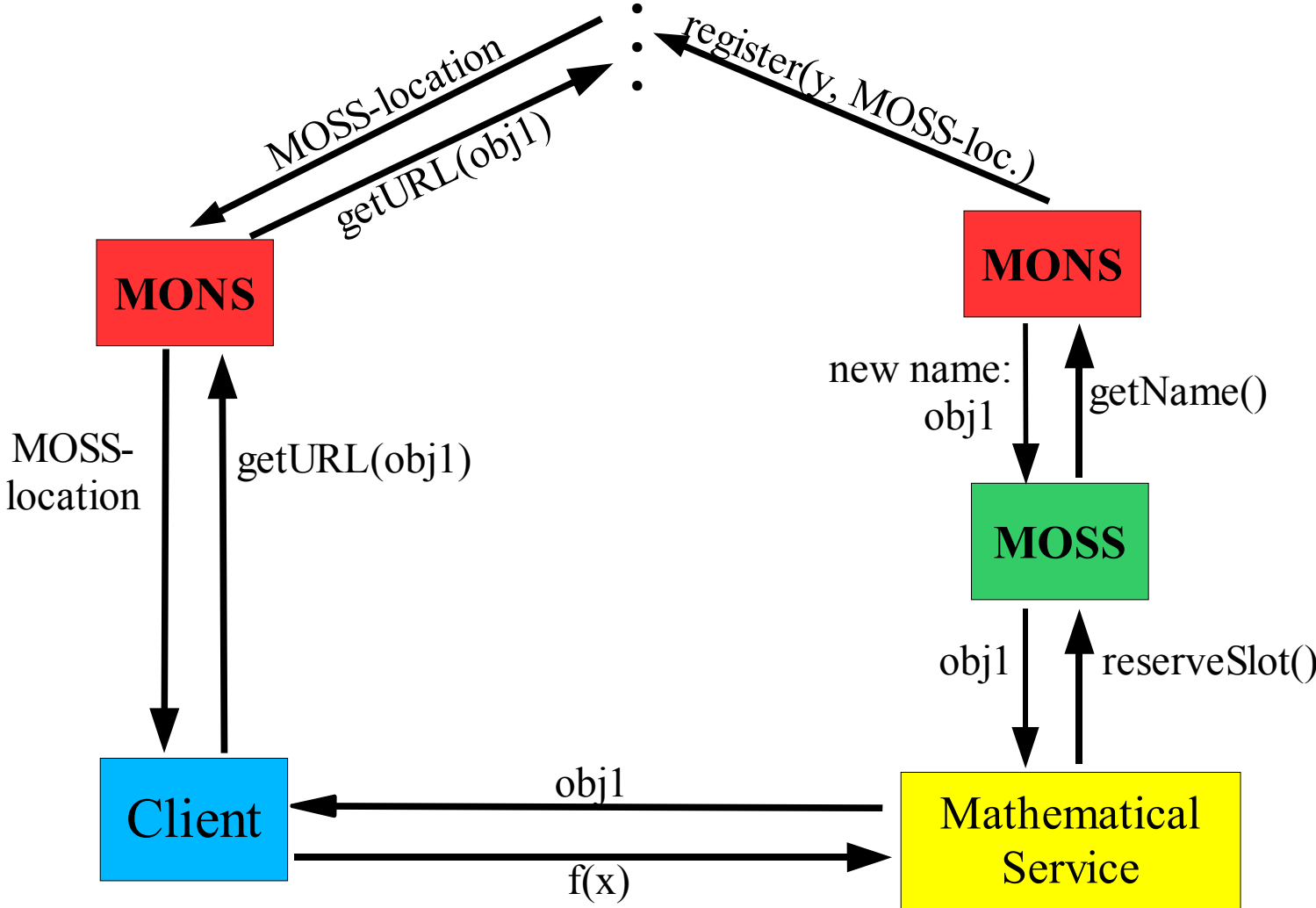


The MONS

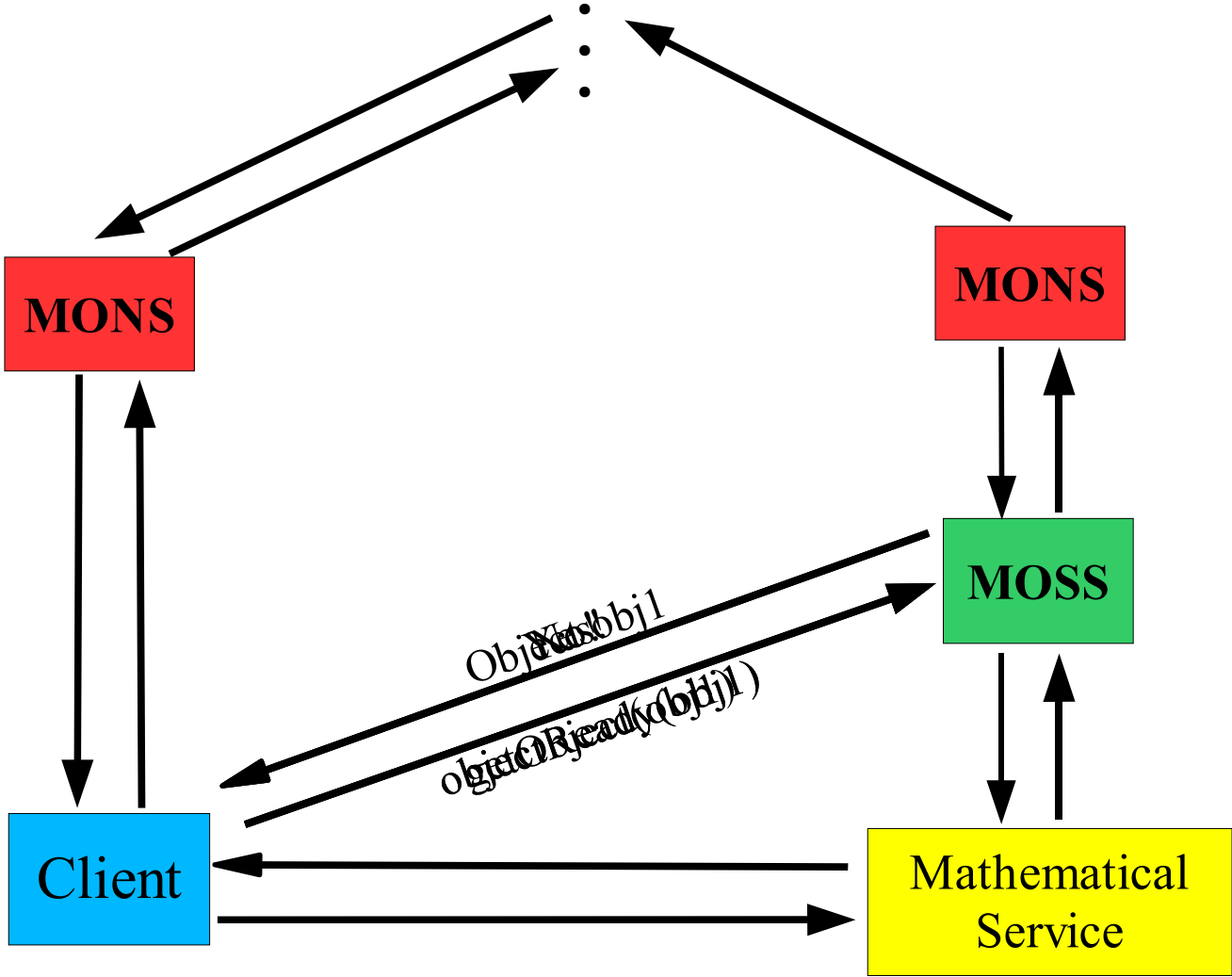


Use cases

Use case 1: Handling of promises

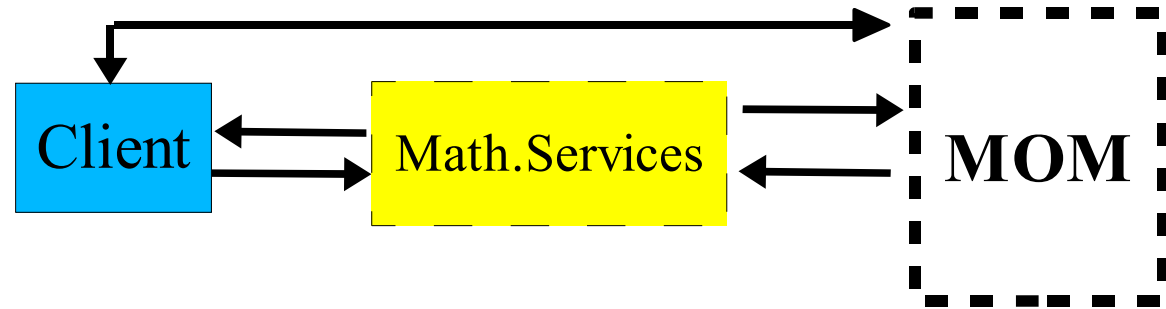


Use case 1: Handling of promises



Use case 2: Execution of a plan

Example of a plan:



$a := A()$



a returned as a reference

$b := B()$



b returned as a reference

$c := C(a, b)$



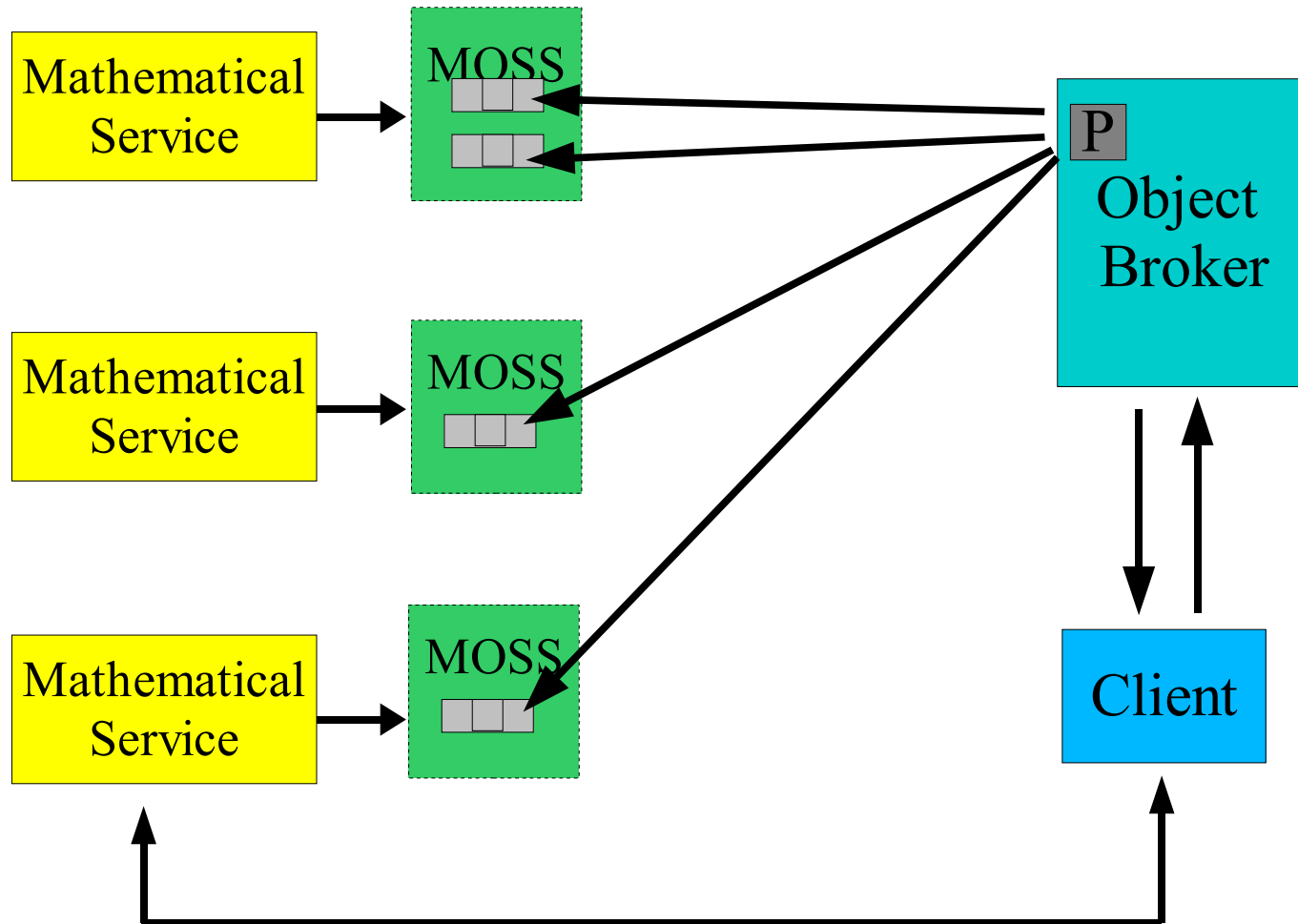
Math. Service requests needed
objects from MOM.

$d := D(a, c)$



- ▶ avoid multiple object transfer
- ▶ avoid EM waiting time

Use case 3: Benchmark sharing



Details on the implementation

Technologies used

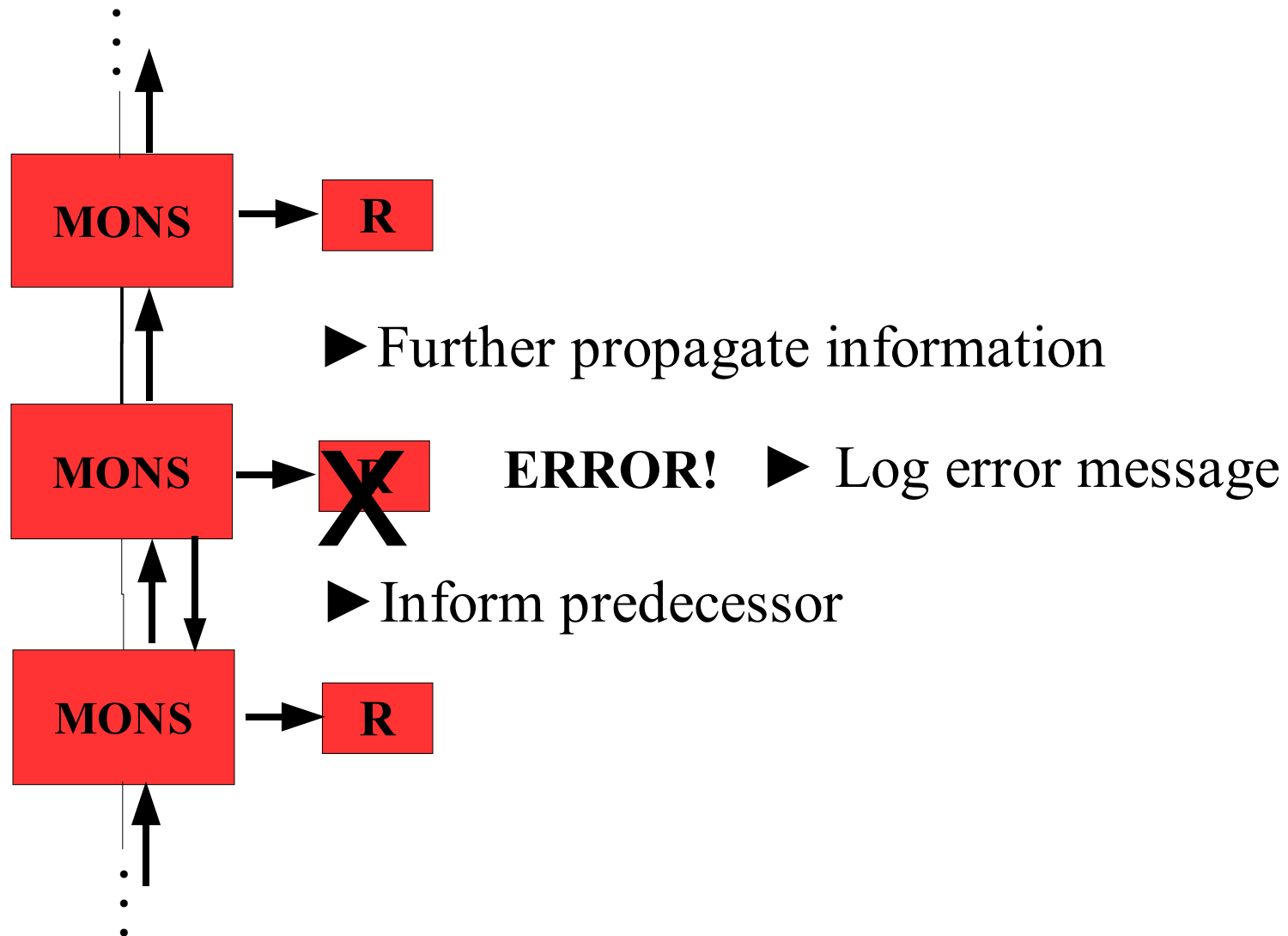
- Web Services (using Tomcat and Axis) to implement MOSS and MONS
- JAXB for handling MOSS database (implemented as an XML file)

Problem handling in the MONS hierarchy

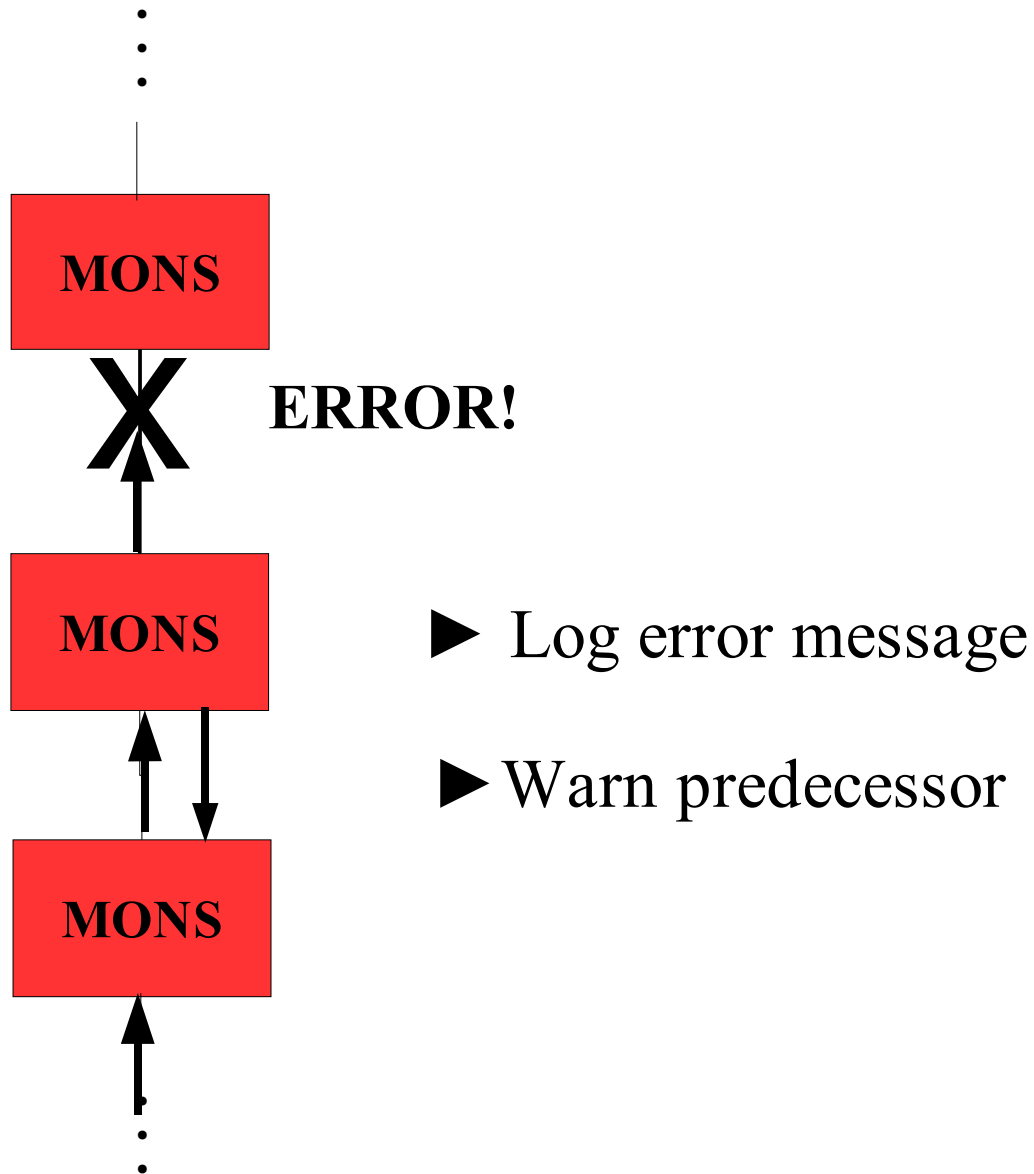
Overall strategy used:

- logging error messages locally for later recovery
- always propagating information further (to/from parent MONS)
- warning direct predecessor about failure

Scenario 1: Repository problem



Scenario 2: Connection problem



Security issues

- ▶ Authentication, secure transmission
 - important, but not yet implemented
 - possibility: use existing modules (e.g. WS Security)

Any questions?