Remote Access to a Symbolic Computation System for Algebraic Topology: a Client-Server Approach

M. Andrés, V. Pascual, A. Romero, J. Rubio

Departamento de Matemáticas y Computación

Universidad de La Rioja

Contents

1. Introduction and Previous Work.

2. First attempts towards distributed (client/server) computing and first problems.

3. Our solution: XML + CORBA.

4. Conclusions and Further Work.

Introduction

EAT (Common Lisp) / KENZO (CLOS)

SYSTEMS FOR SYMBOLIC COMPUTATION IN ALGEBRAIC TOPOLOGY

$H_5(\Omega^3 \text{Moore}(Z_2, 4))$ ⟶

$(Z_2)^5$ ⟶

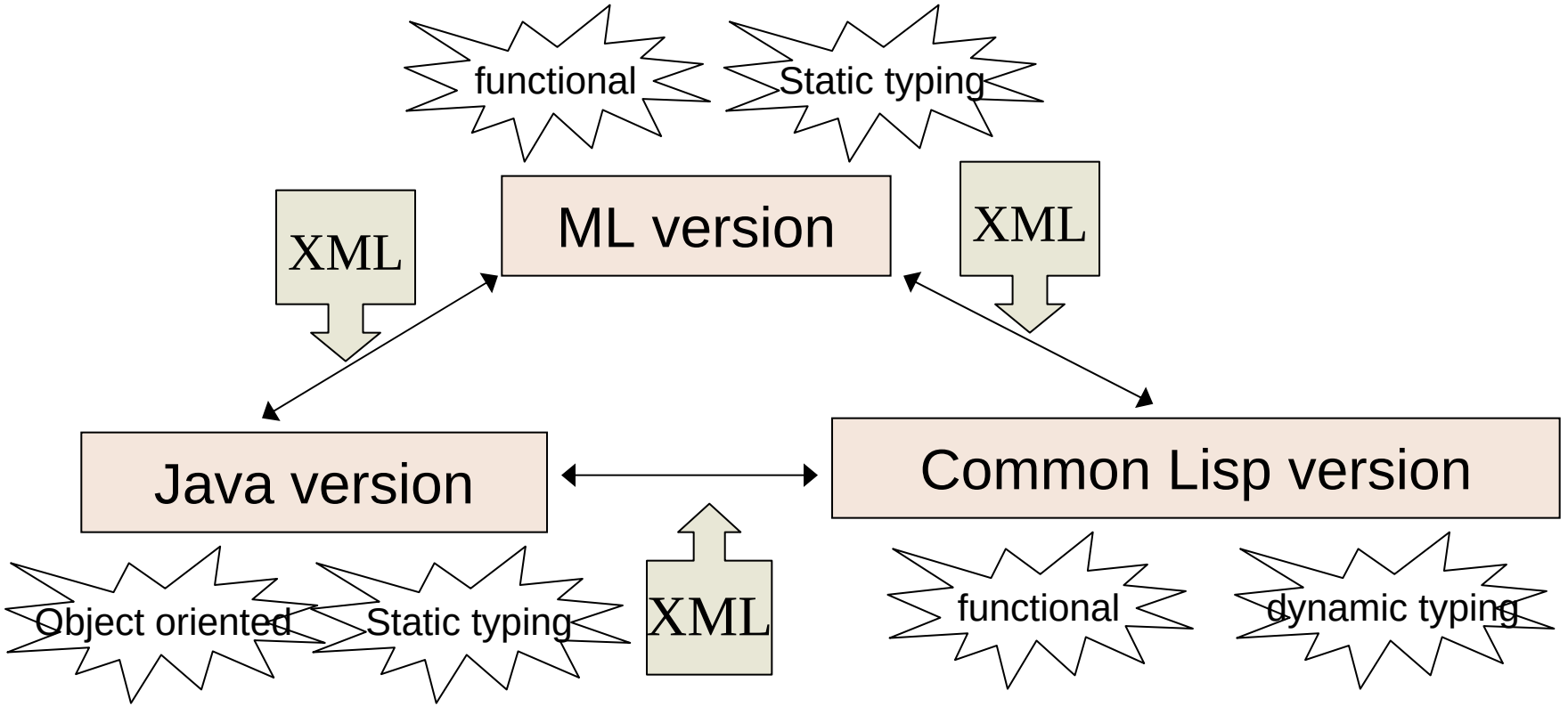
```
>(homology (loop-space (moore 2  4) 3) 5)

homology  in dimension 5:
        Component Z/2Z
        Component Z/2Z
        Component Z/2Z
        Component Z/2Z
        Component Z/2Z
```

- Intensive use of functional programming.
- Discovery of unknown results.
- Usability??
- Remote access??

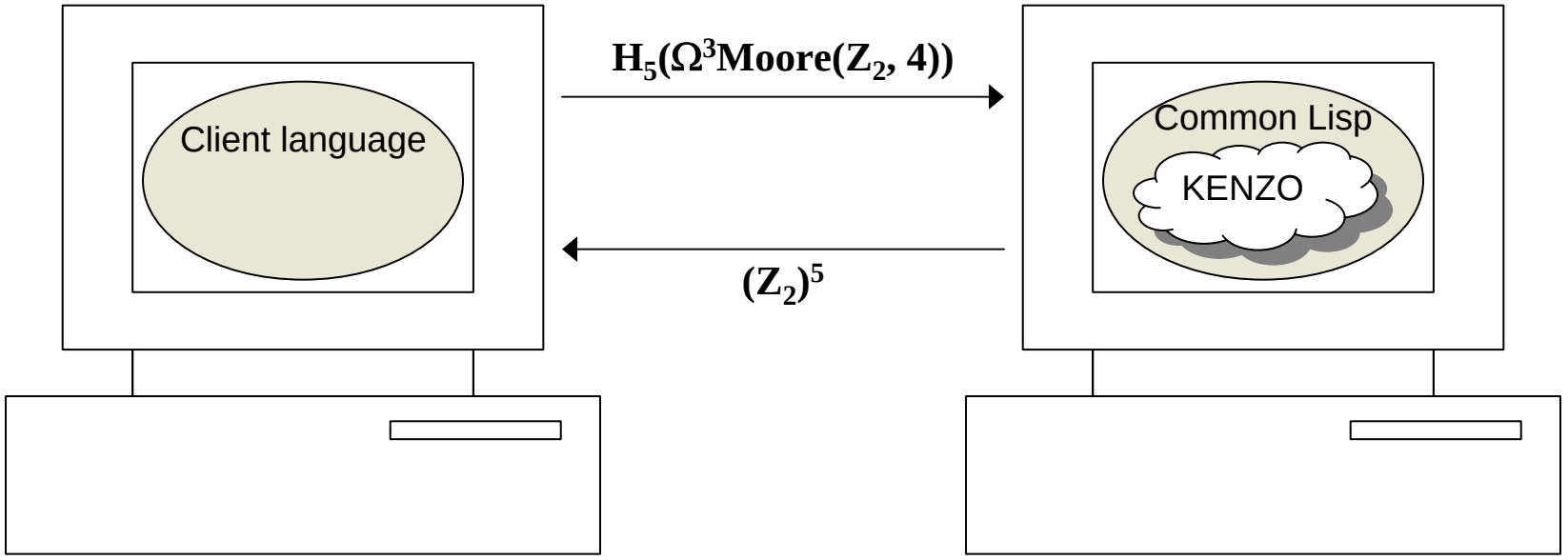
Previous Work

- Reconstruct (part of) them in Java. As an intermediate step: ML.
- To interoperate between the three versions: XML (extending MathML)



only in a local non-distributed way

Goal: distributed computing



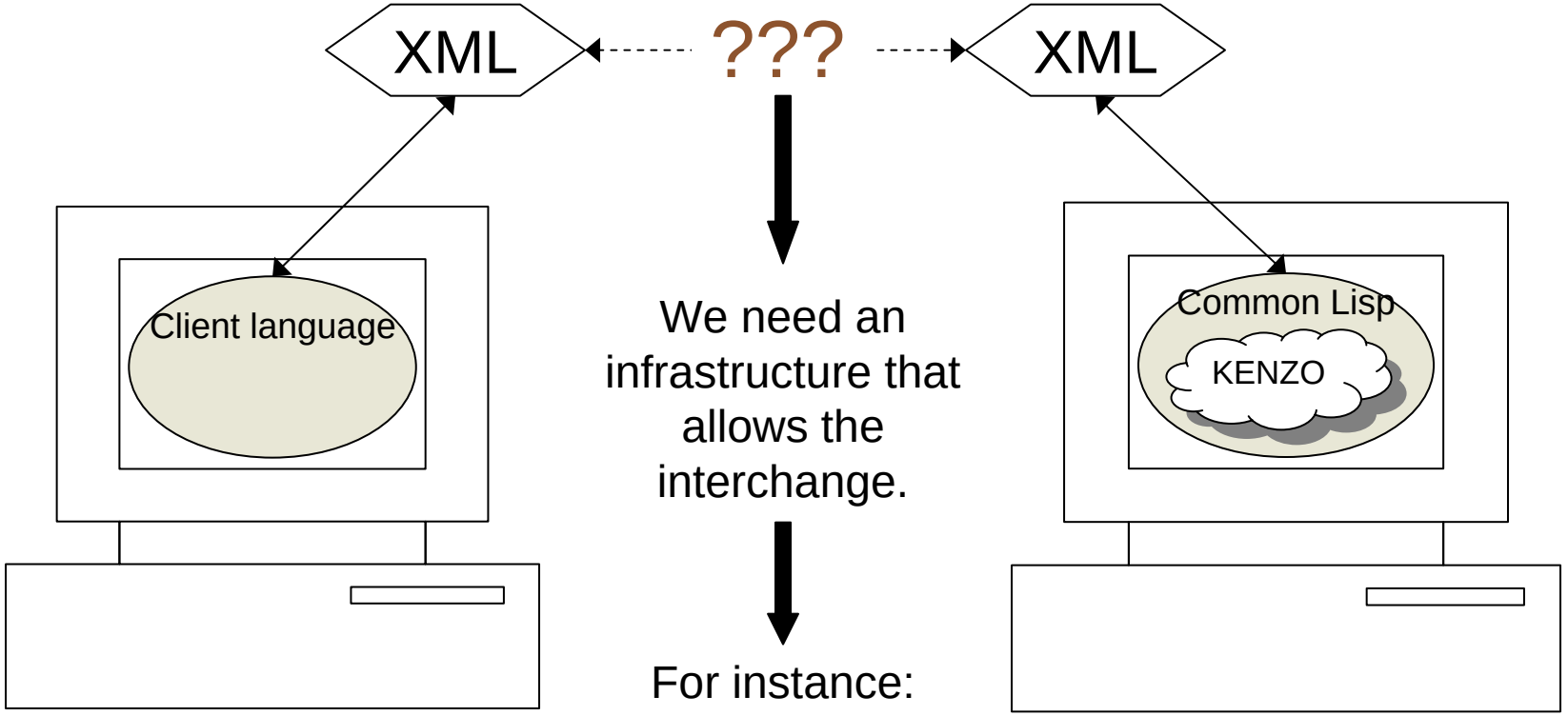
CLIENT SERVER

Important features of EAT and Kenzo to rebuild them as distributed systems:

- Very time and space consuming.
- Written in Common Lisp.
- Higher order functional programming (infinite data structures).
- Organized in two layers of data structures.

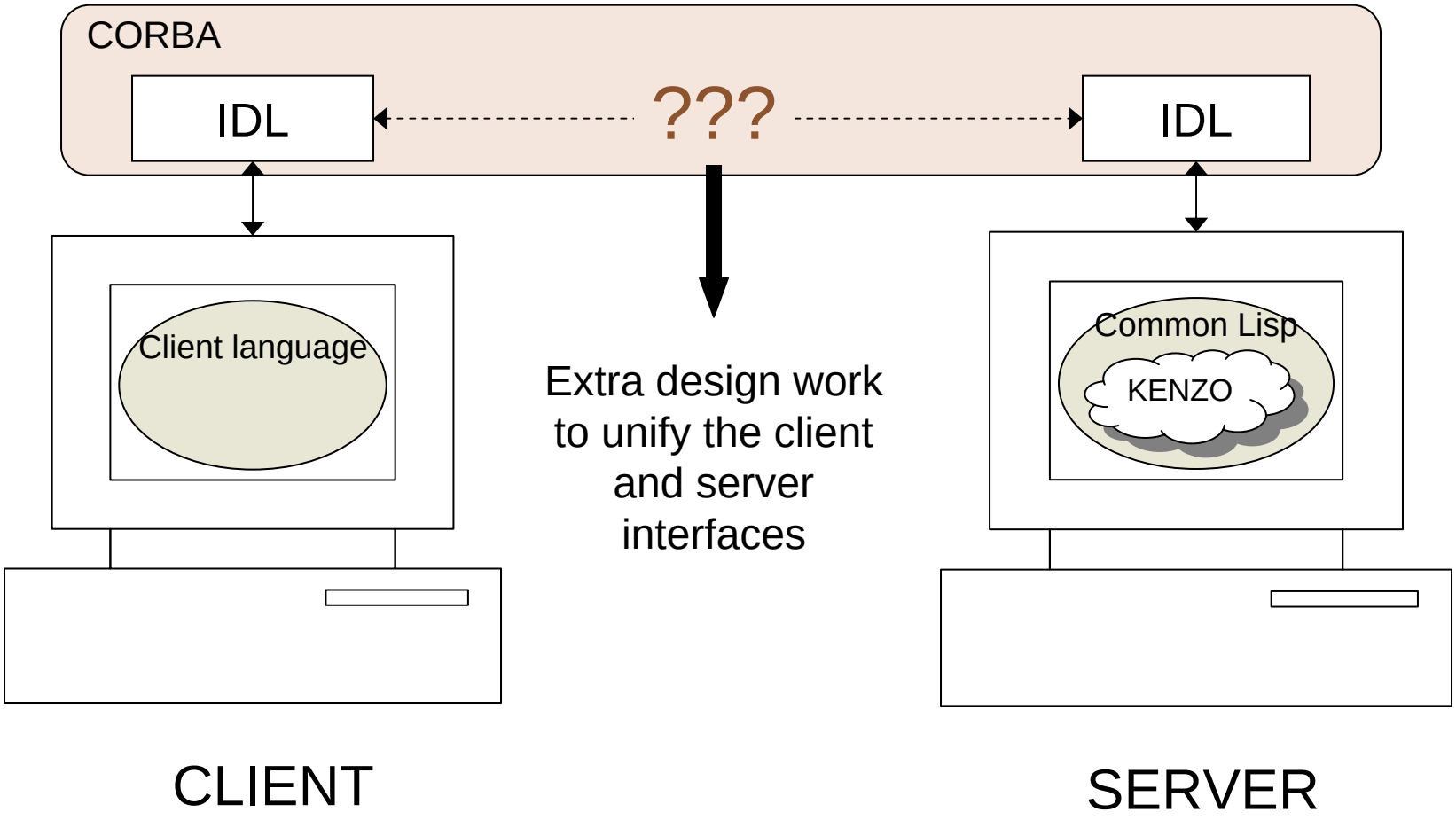
First attempts, first problems

1) Use of XML



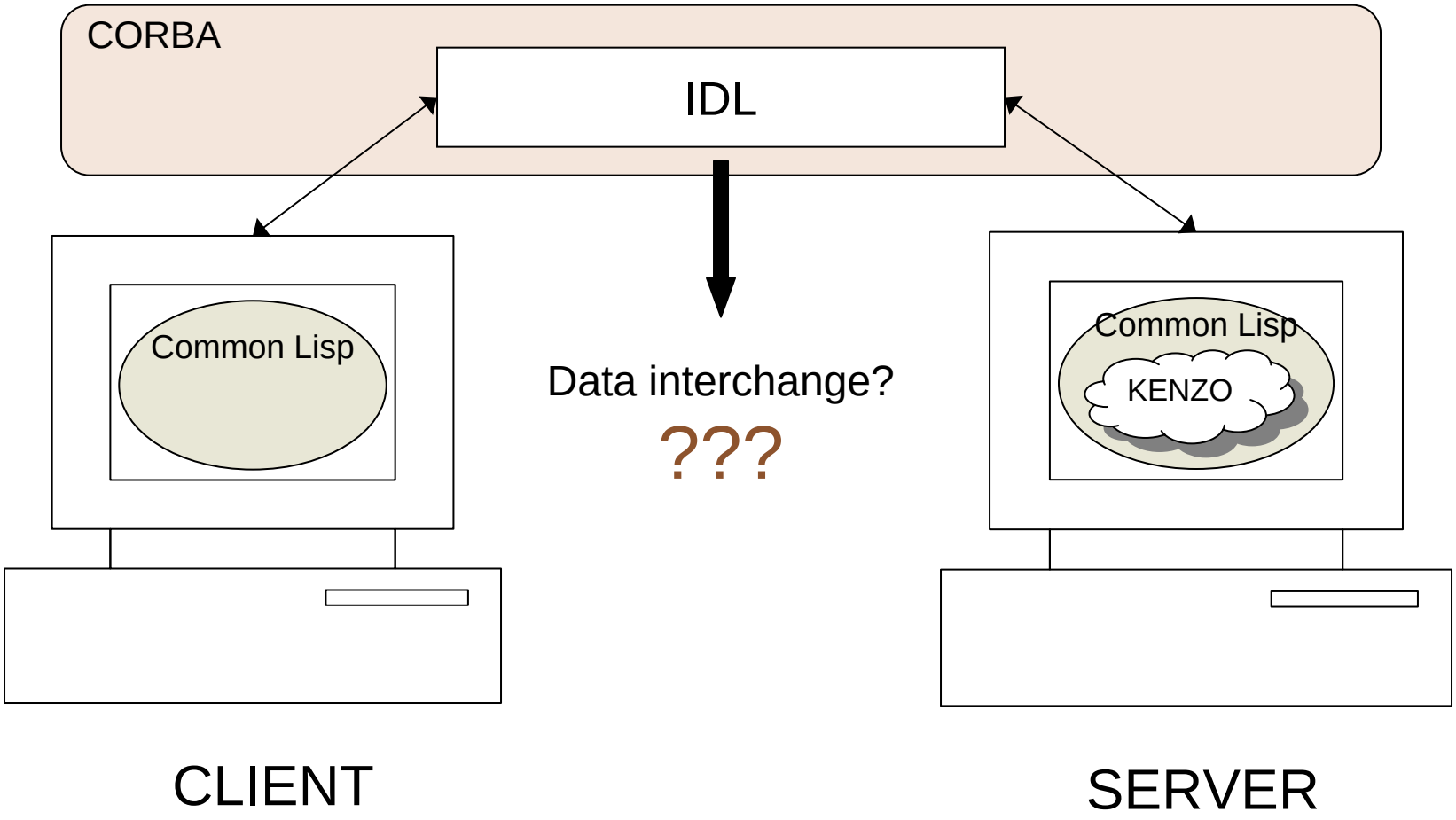
First attempts, first problems

2) CORBA (manage a solution to the problem of distributed interoperability)



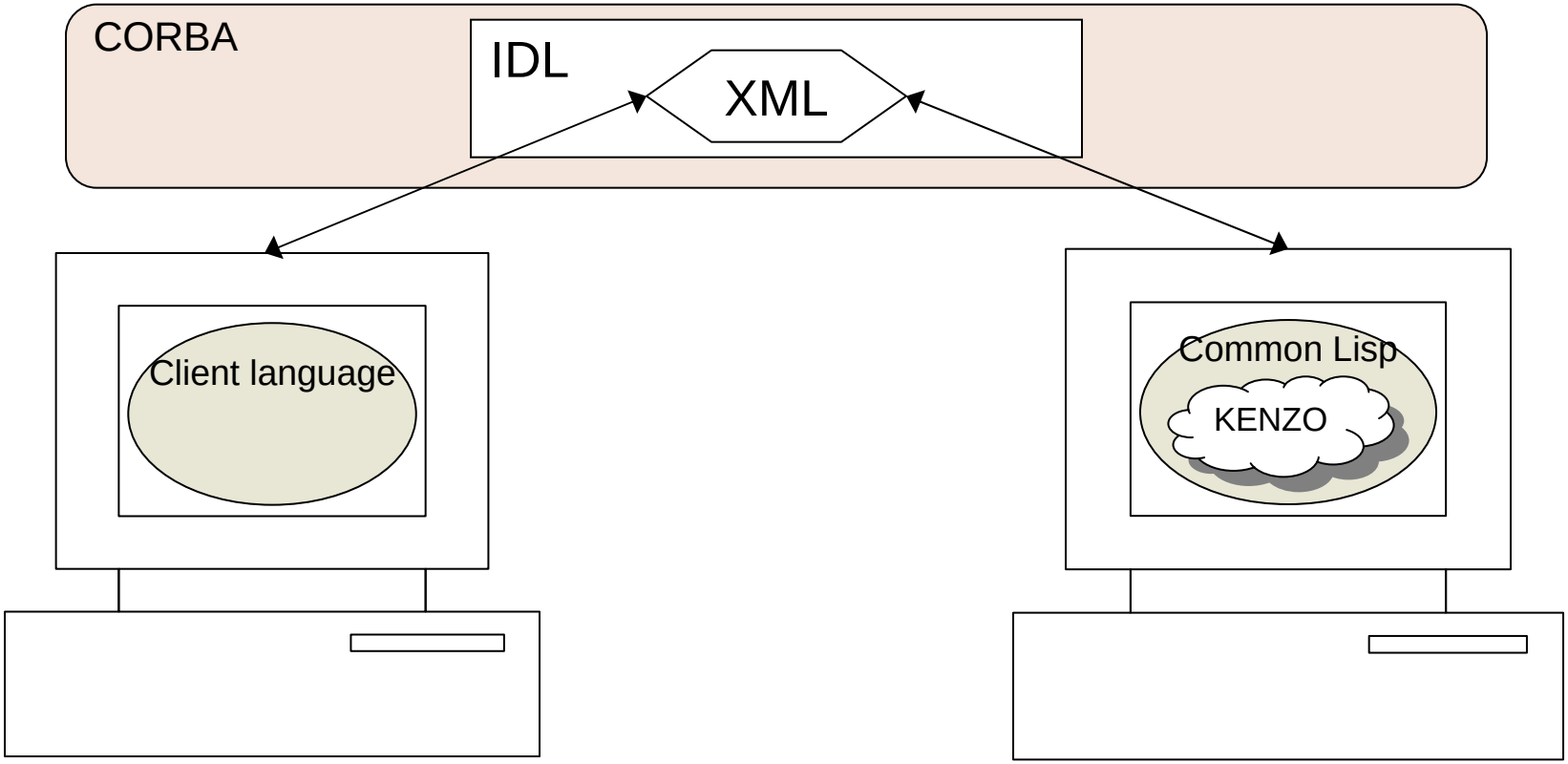
First attempts, first problems

3) CORBA in an homogeneous environment



Our solution: XML + CORBA

XML as the necessary “glue” to solve the problem of data interchange.



CLIENT

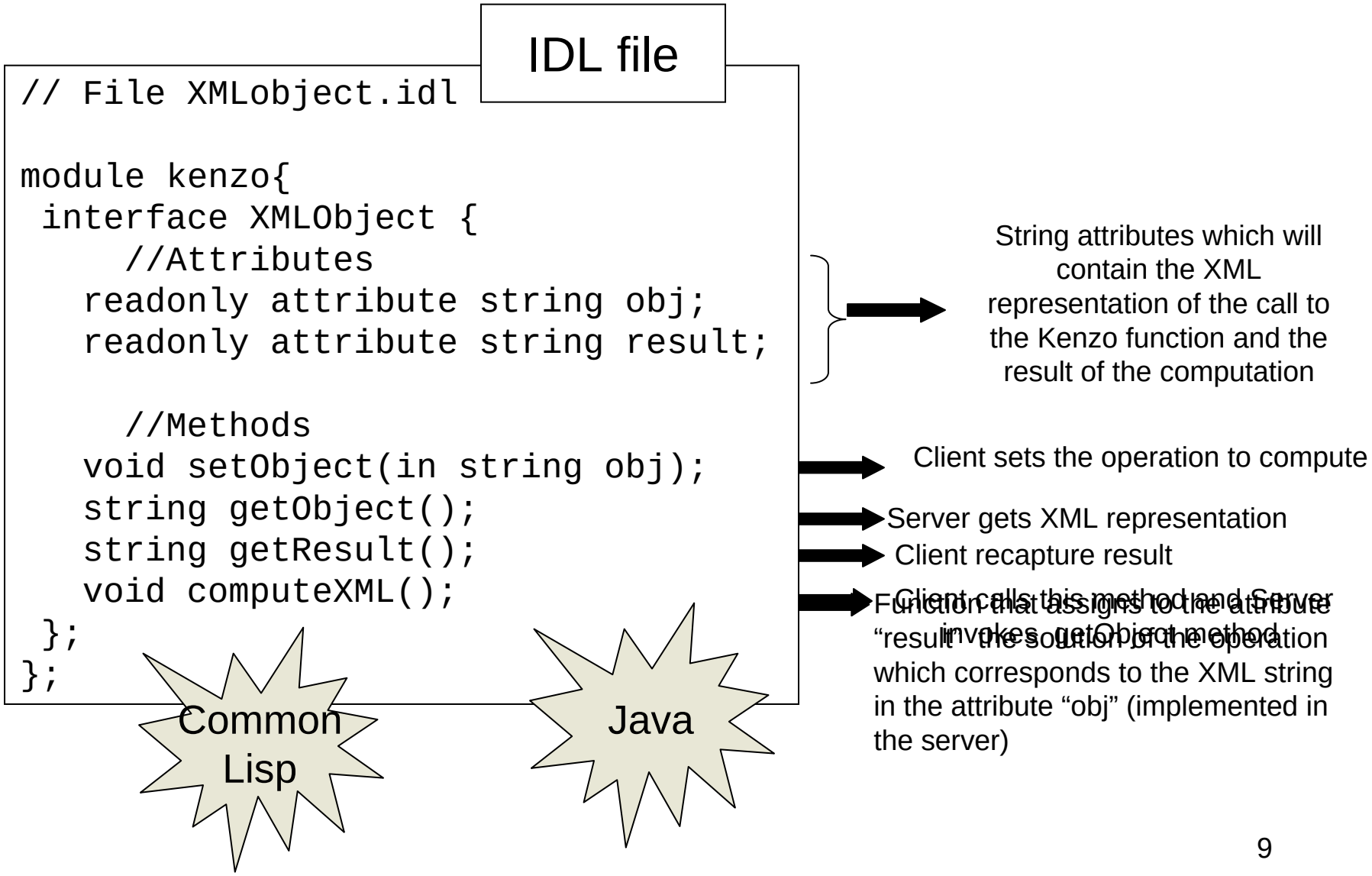
-IDL unification

SERVER

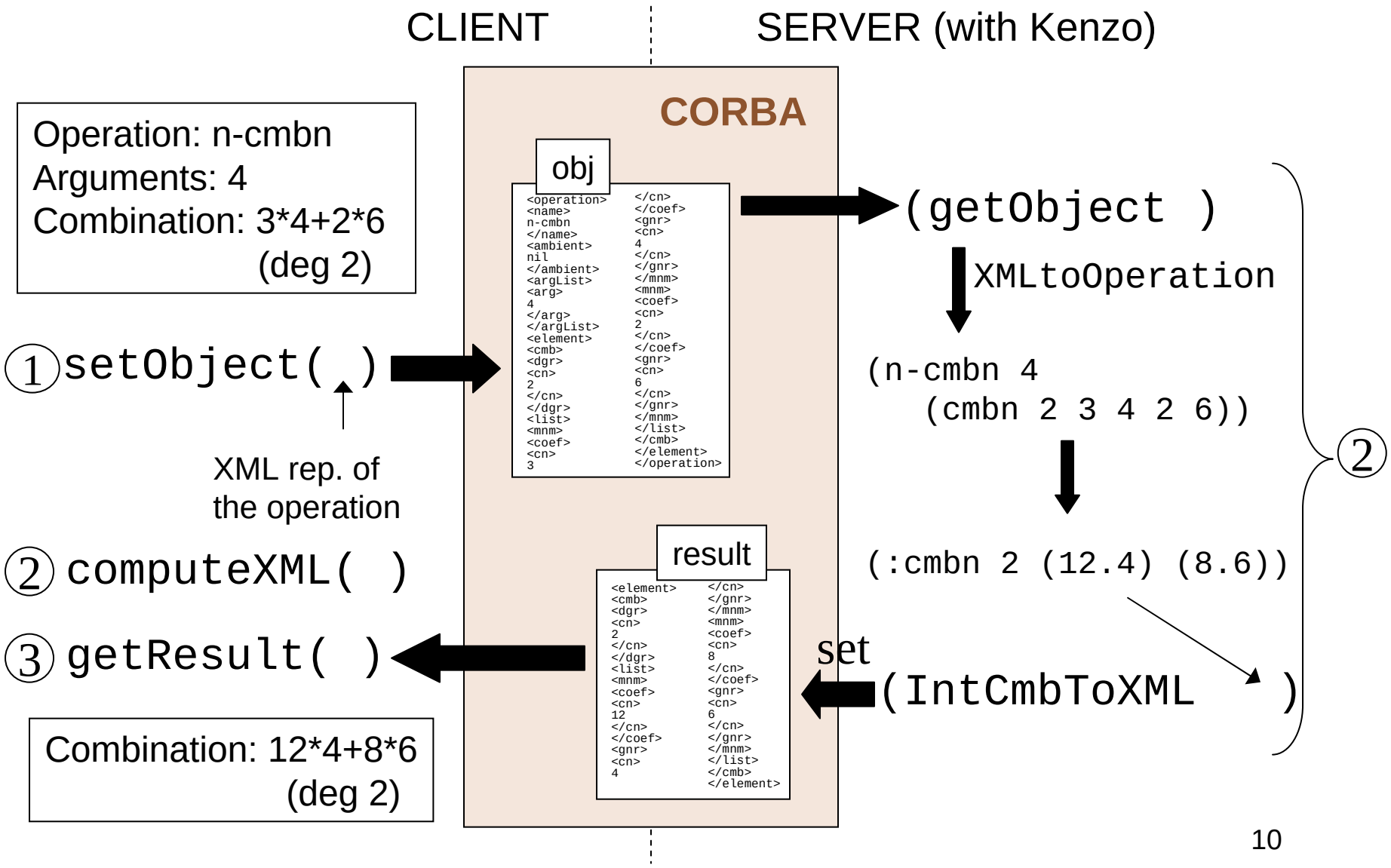
-same interface for all types of data

-reuse previous prototypes (ML and JAVA)

Our solution: XML + CORBA



Example of client/server interoperability using CORBA/XML



XML-based interchange format

Elements (data)

```
<element>
<cmb>
<dgr>
<cn>
2
</cn>
</dgr>
<list>
<mnm>
<coef>
<cn>
12
</cn>
</coef>
<gnr>
<cn>
4
</cn>
<cn>
8
</cn>
</coef>
<gnr>
<cn>
6
</cn>
</gnr>
</mnm>
</list>
</cmb>
</element>
```

Combination 12*4+8*6
(deg 2)

Operations

```
<operation>
<name>
n-cmbn
</name>
<ambient>
nil
</ambient>
<argList>
<arg>
4
</arg>
</argList>
<element>
<cmb>
<dgr>
<cn>
2
</cn>
</dgr>
<list>
<mnm>
<coef>
<cn>
3
</cn>
</coef>
<gnr>
<cn>
4
</cn>
</gnr>
</mnm>
<mnm>
<coef>
<cn>
2
</cn>
</coef>
<gnr>
<cn>
6
</cn>
</gnr>
</mnm>
</list>
</cmb>
</element>
</operation>
```

Operation: n-cmbn
Arguments: 4
Combination: 3*4+2*6 (deg 2)

Conclusions

- From local interoperability with XML to client-server XML-interoperability (based in CORBA) in a local net (Goal achieved: reusing the software previously created).
- The XML format that allows the interchange is a proper extension of MathML.
- (Corba-based) Remote access in a local net (not really distributed computing).
- Clearly, a necessary step towards distributed computing.

Further work

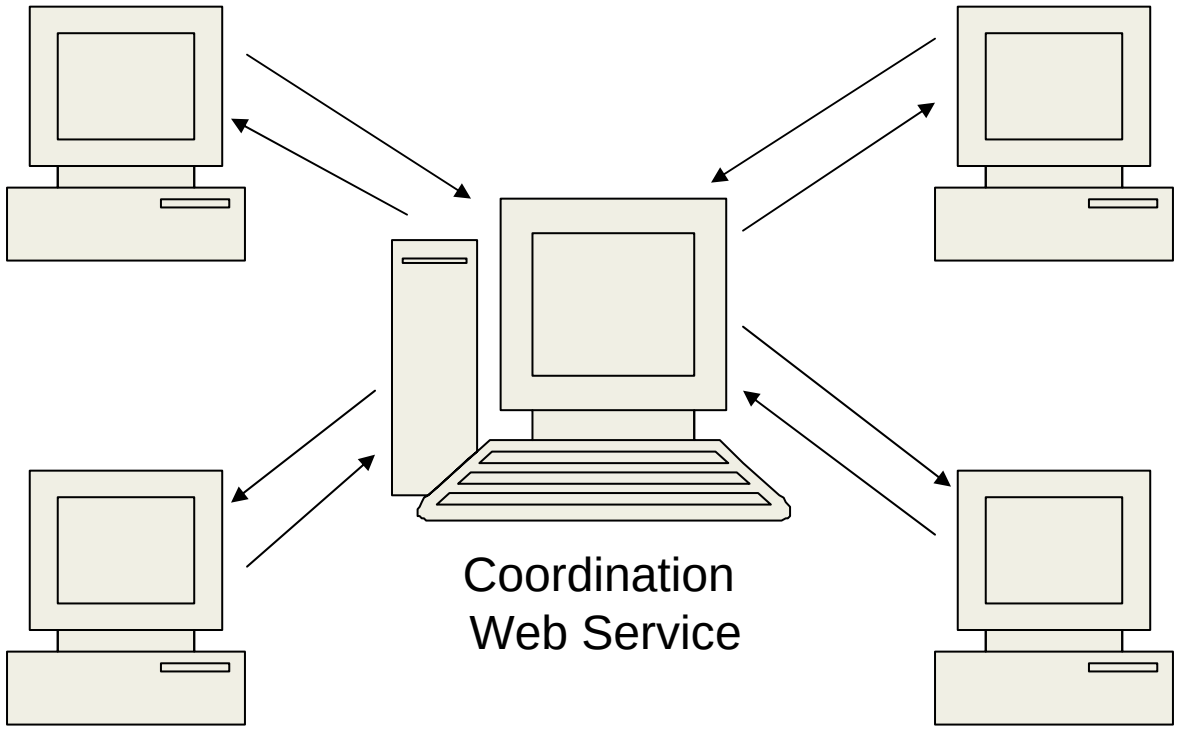
- Now we manage a client/server interoperability working in a local net due to CORBA
- The extension to the Internet is a simple technical step
- Interest in some kind of asynchronicity
- Subscription system ??

Towards Web Services

Further work

Web Services to get real distributed computing.

Infraestructure: several servers cooperating in the same calculation.



That is all !

Thank you very much !!!

Questions??