

28th EUROMICRO CONFERENCE

Dortmund September 4th-6th,2002

COMPONENT-BASED SOFTWARE ENGINEERING

**Local Components and Reuse of
Legacy Code in the CORBA
Component Model**

Egon Teiniker, Stefan Mitterdorfer, Christian Kreiner, Zsolt Kovacs,
Reinhold Weiss

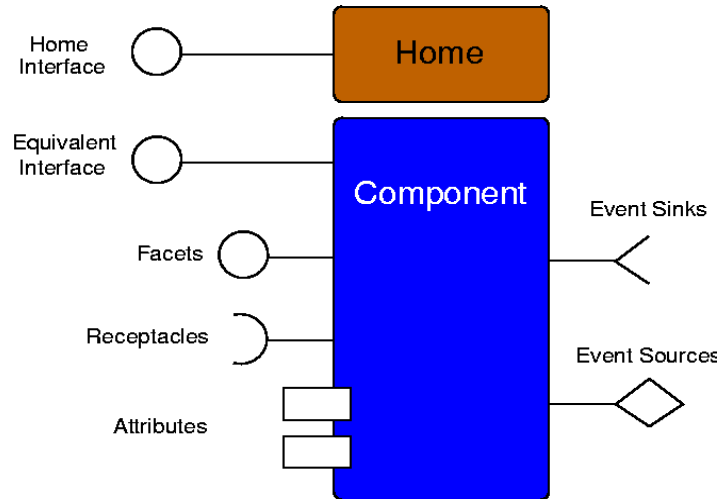
**Institute for Technical Informatics
Graz University of Technology
AUSTRIA**

Overview

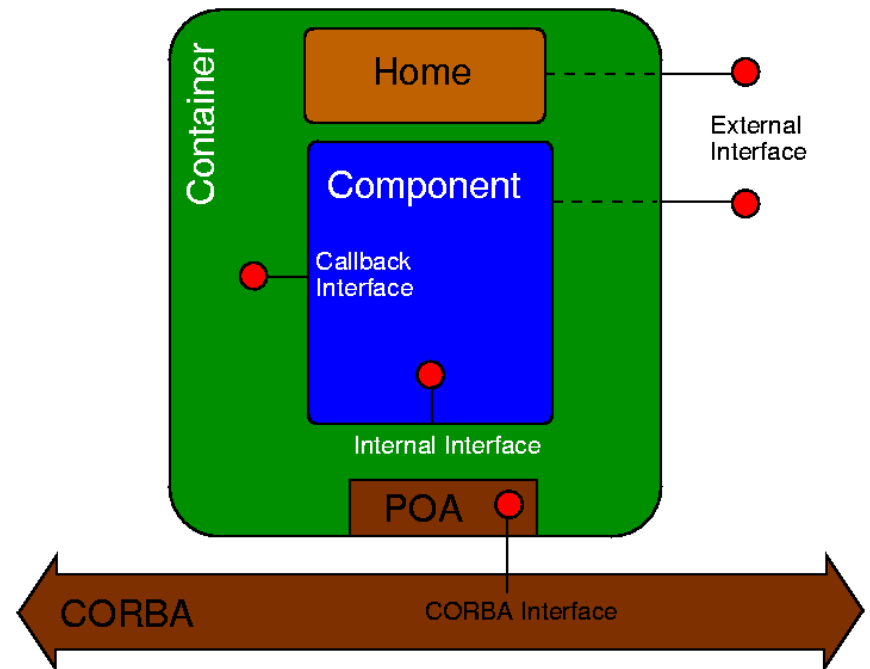
- **CORBA Component Model**
- **Development Requirements**
- **Related Work**
- **Local Component Adapter Concept**
- **Remote vs. Local Components**
- **Reuse of Legacy Code**
- **Component Development Process**
- **Experimental Results**
- **Conclusion and Future Work**

CORBA Component Model

CORBA Component

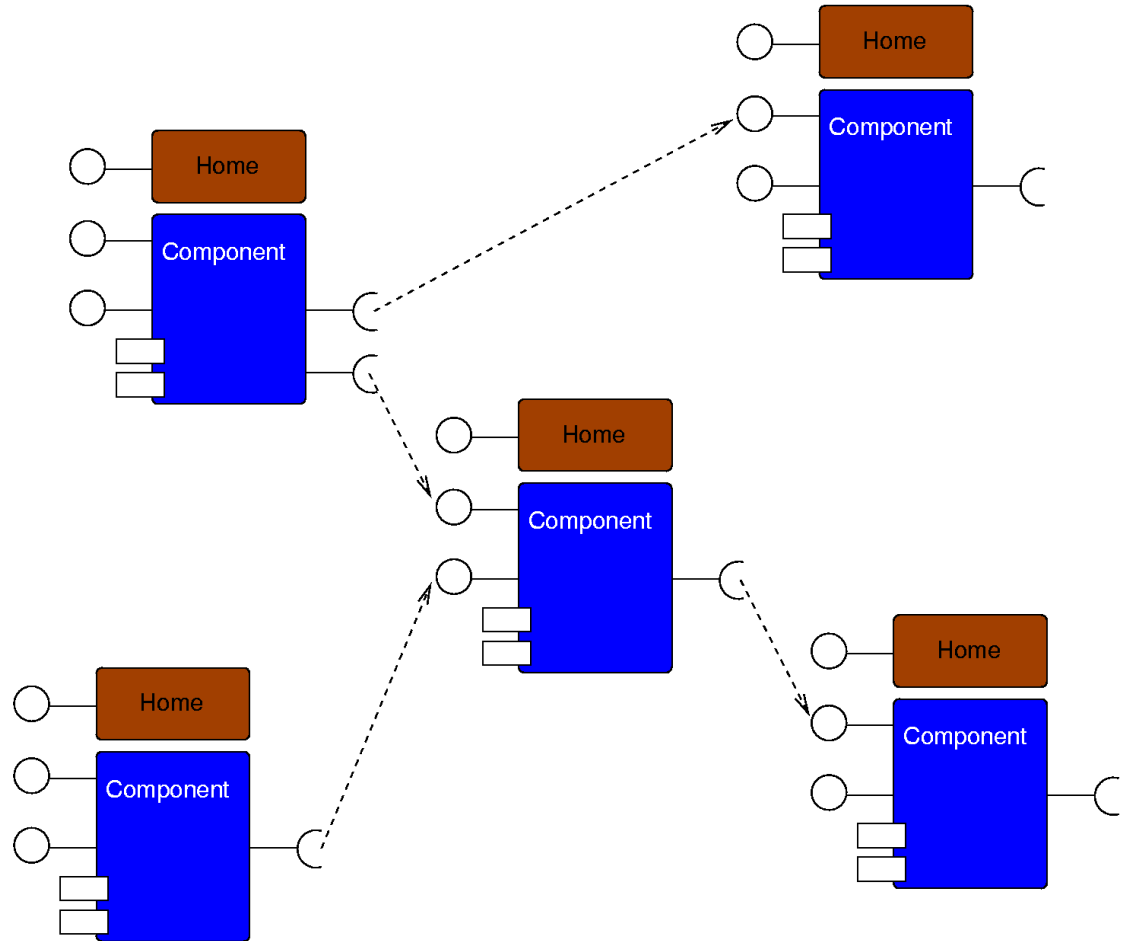


Component Container



CORBA Component Model

CORBA Component Assembly



Development Requirements

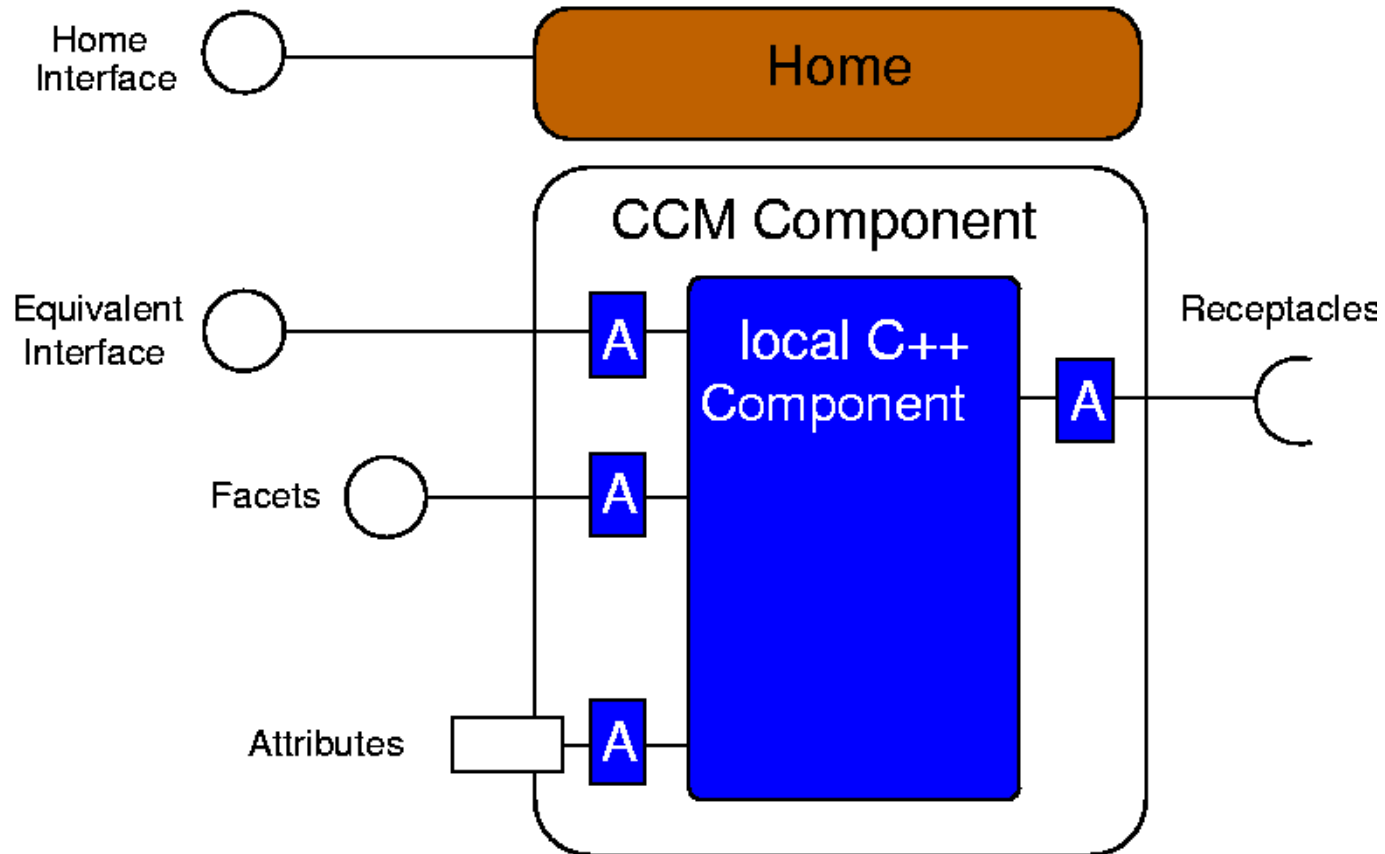
Problems with CORBA Component Model:

- **CORBA is complicated**
 - Business logic
 - Legacy code
- **Components in the same address space**
 - Performance

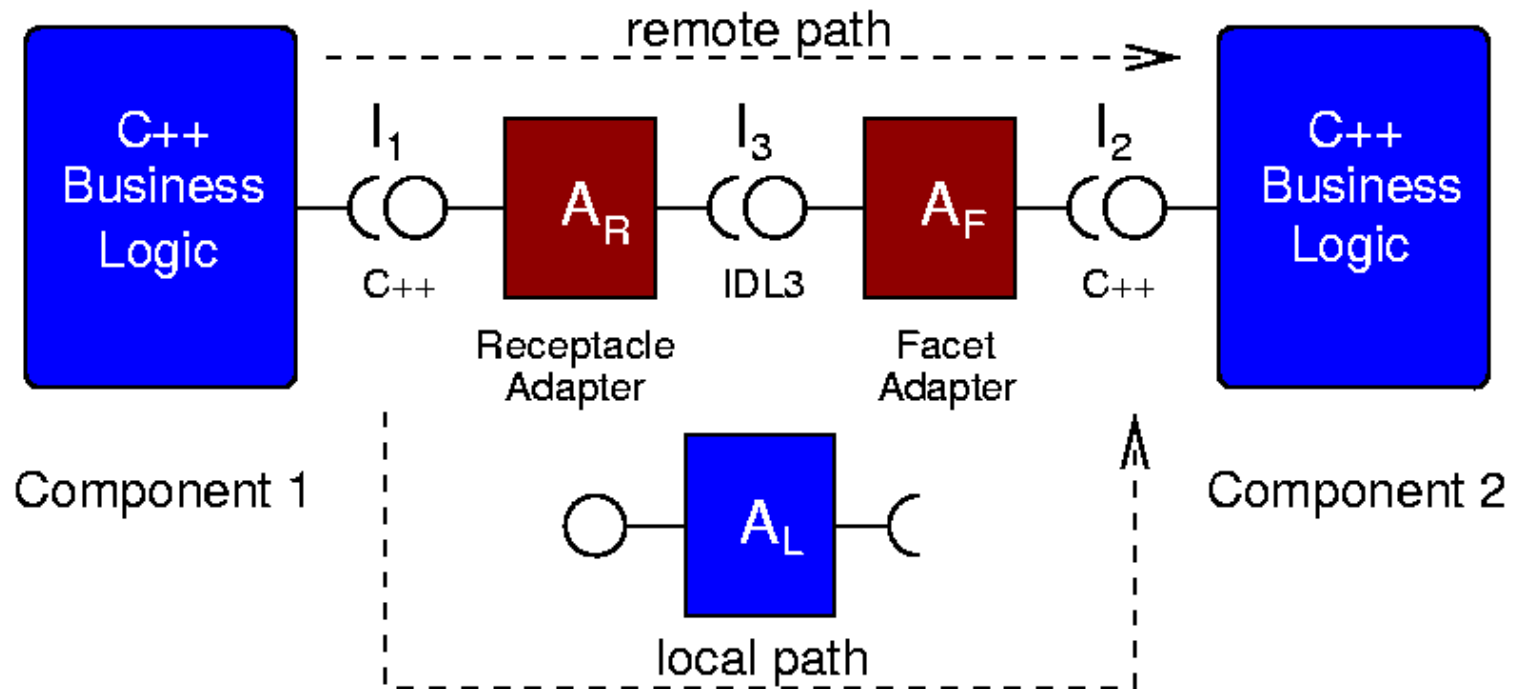
Related Work

- **Local Enterprise JavaBeans**
 - since EJB 2.0
- **CORBA Collocation**
 - Standard Collocation
 - Direct Collocation
- **CCM Language Mapping**
 - Local CORBA interfaces

Local Component Adapter Concept

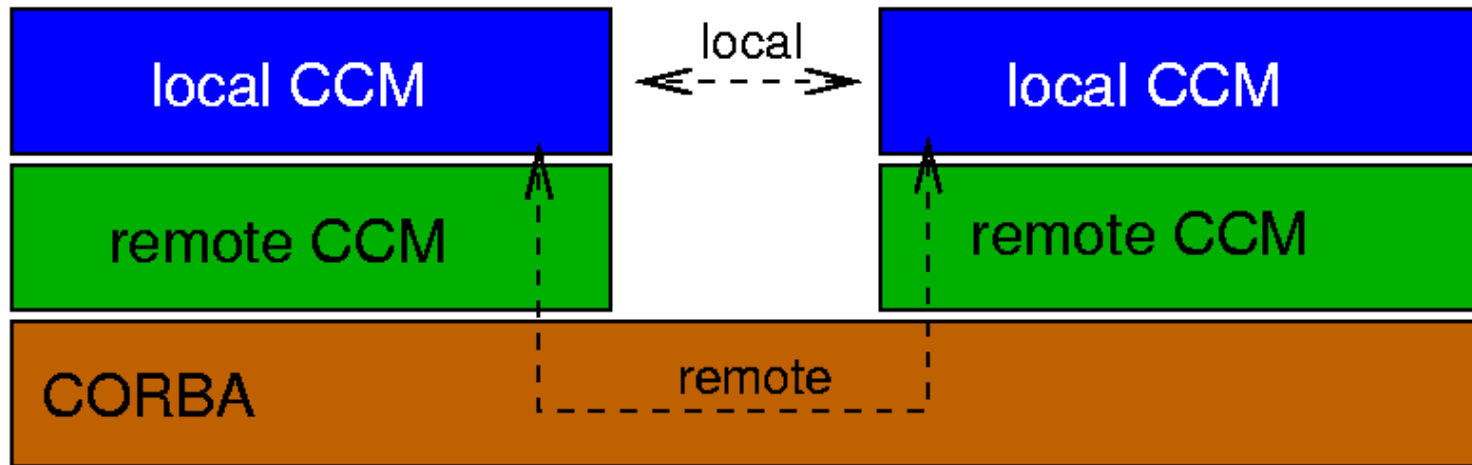


Local Component Adapter Concept

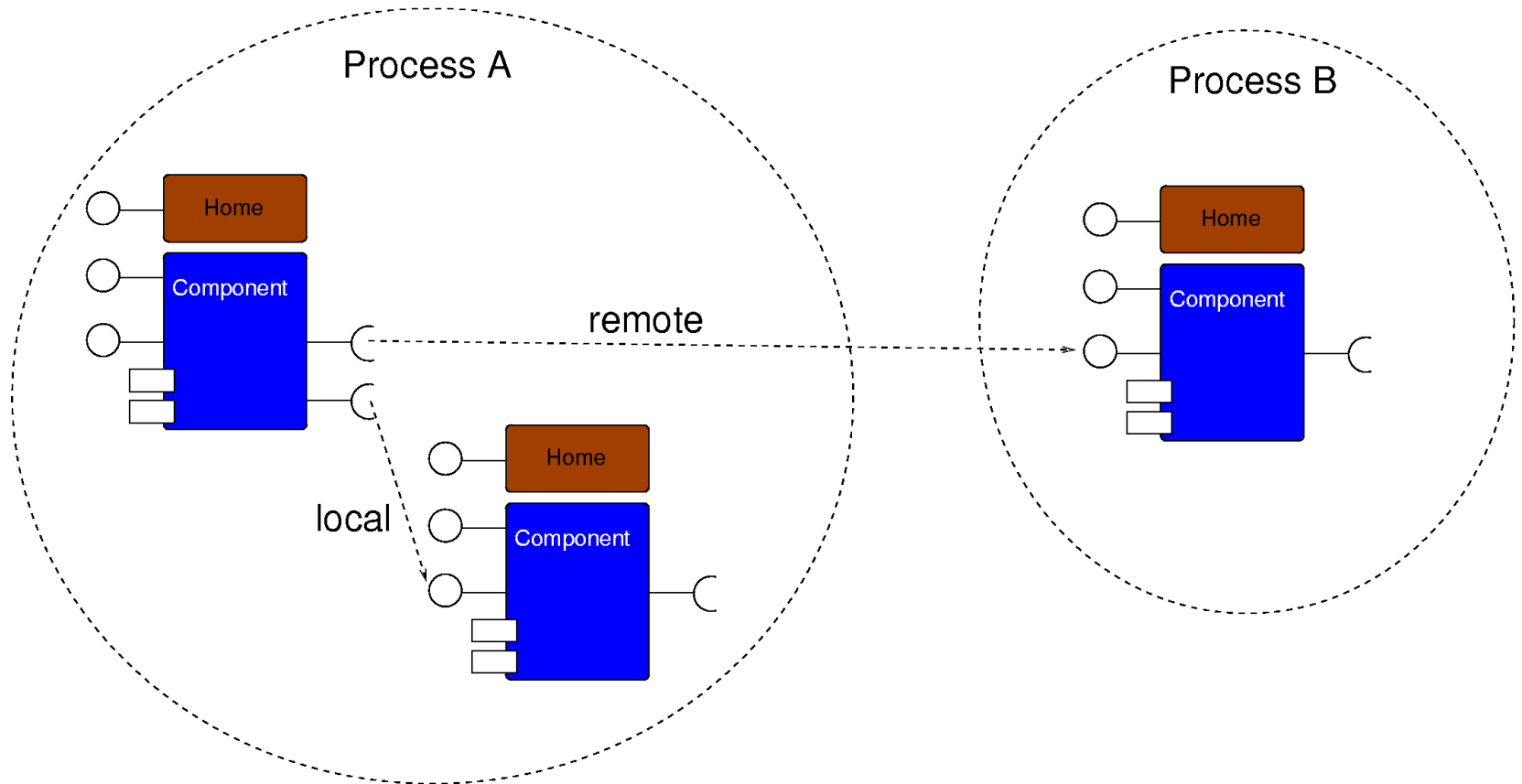


Remote vs. Local Components

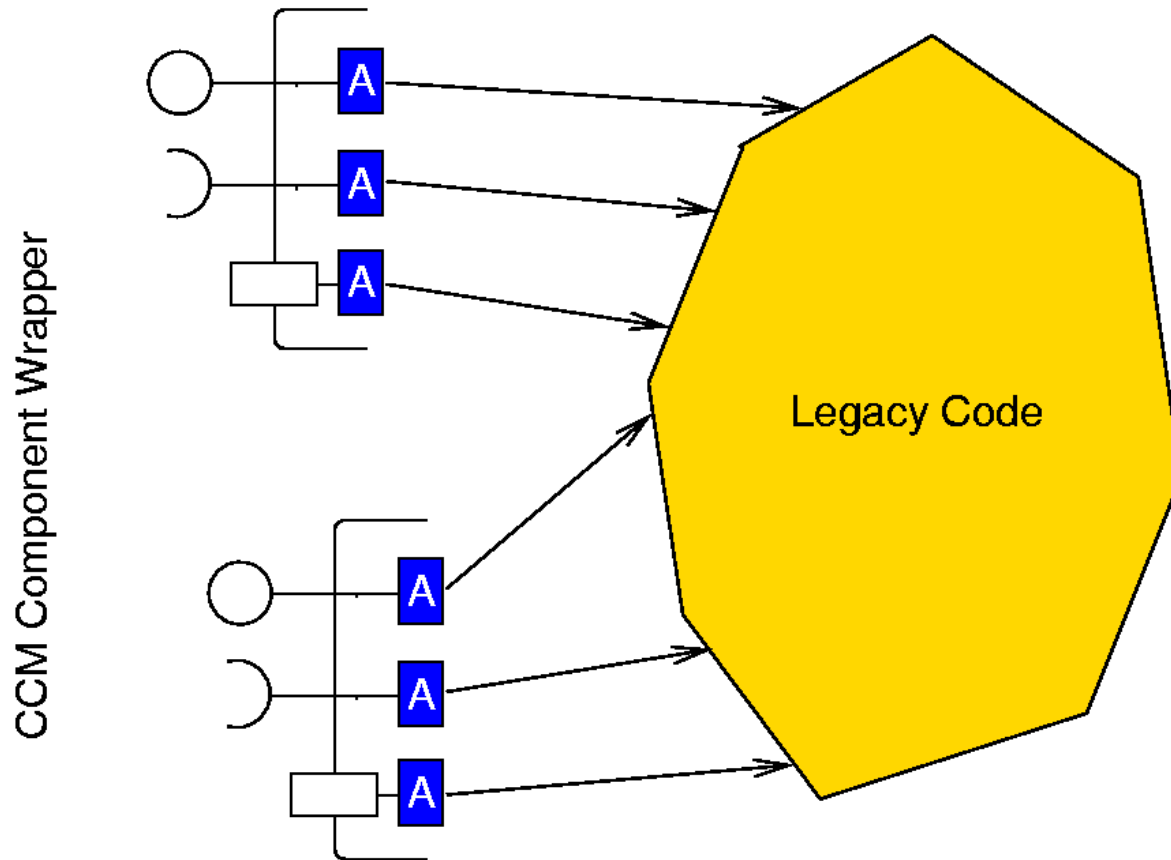
IDL3 / C++



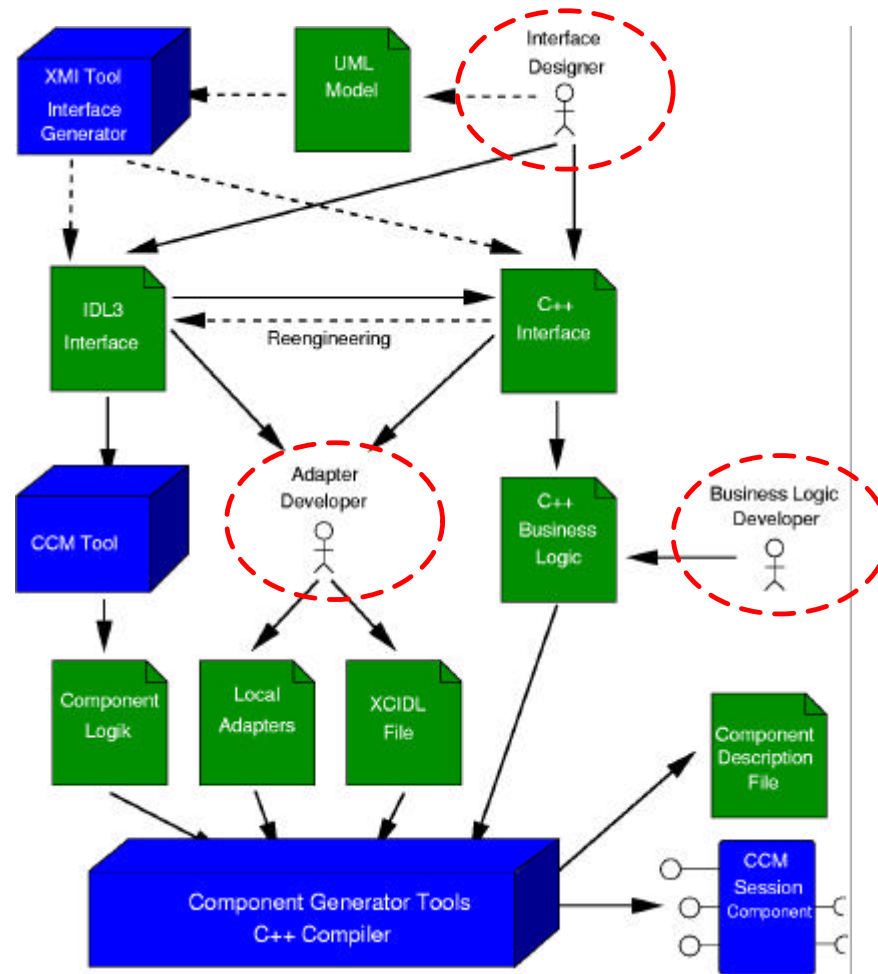
Remote vs. Local Components



Reuse of Legacy Code

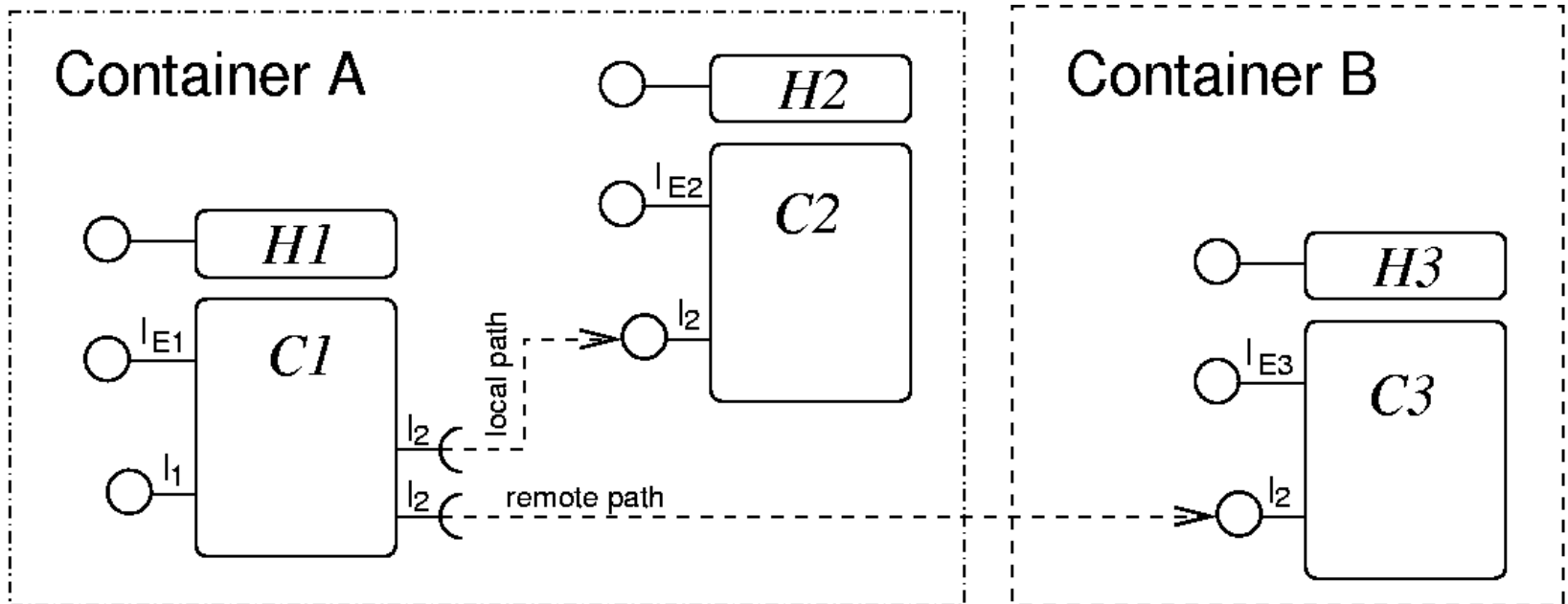


Component Development Process



Experimental Results

Example Assembly:



Experimental Results

```
typedef sequence<long> LongSeq;  
interface I2 {  
    long f1(in long a);  
    long f2(in string a);  
    long f3(in LongSeq s);  
}
```

	remote [ms]	local [ms]
f1()	2824	36
f2()	4478	32
f3()	11758	32

1.000.000 Requests on a Pentium III 700MHz

Conclusion and Future Work

- **Current status:**

- Local component adapter concept
 - no CORBA in business code
 - easy reuse of legacy code
- Assemblies of local and remote components
 - local components

- **Next steps:**

- Embedding scripting languages in CCM
- Integration of entity concept
- Integration of event handling