

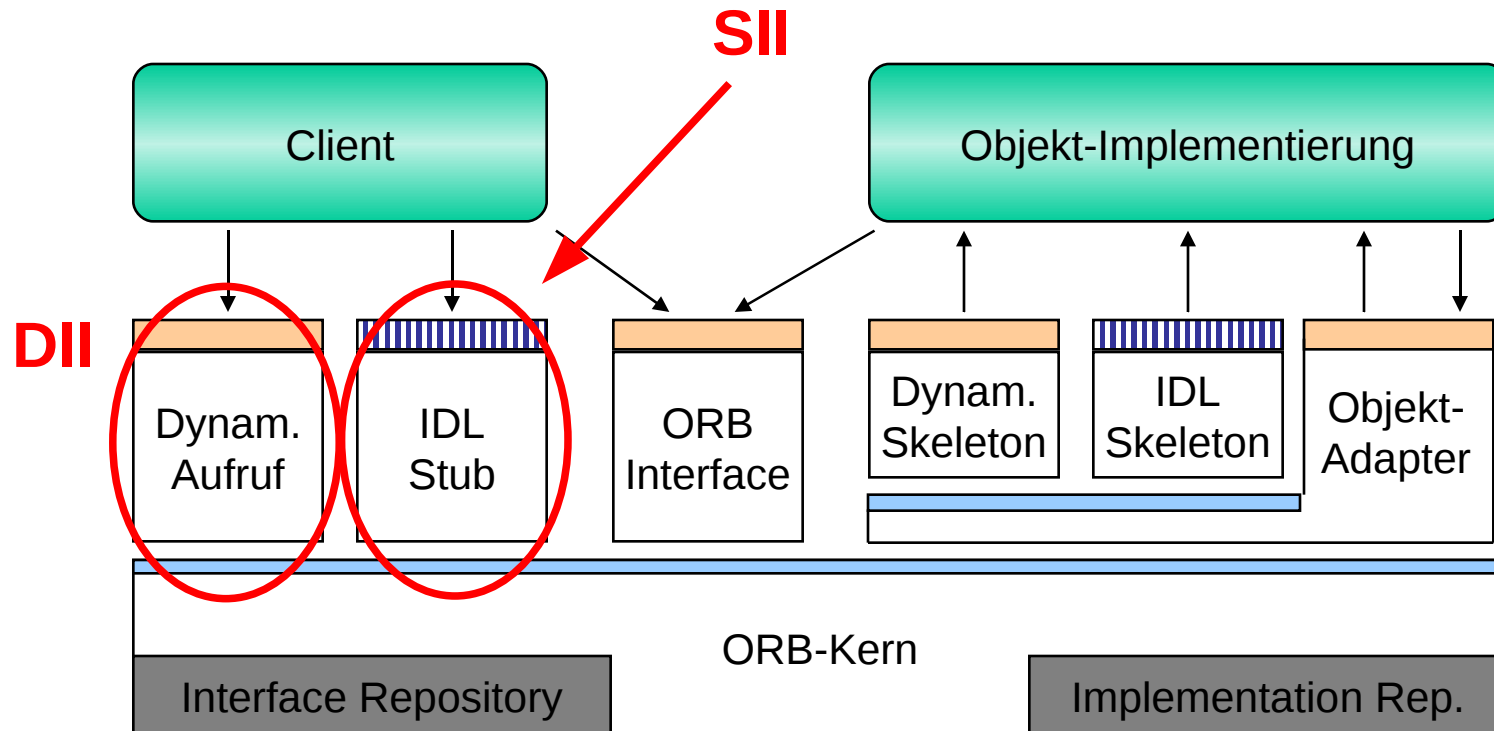
# Internetanwendungstechnik (Übung)

## CORBA

S. Bissell, G. Mühl

Technische Universität Berlin  
Fakultät IV – Elektrotechnik und Informatik  
Kommunikations- und Betriebssysteme (KBS)  
Einsteinufer 17, Sekr. EN6, 10587 Berlin

# CORBA



- standardisiert
- intern
- Anwendung

# SII vs. DII

## Static Invocation Interface (SII)

- > einfacher zu benutzen
- > durch IDL Compiler
- > zur Compilezeit festgelegt
- > type checking

## Dynamic Invocation Interface (DII)

- > aufwendiger zu schreiben
- > durch Programmierer
- > zur Laufzeit
- > manuelles type casting

# Asynchrone Methodenaufrufe

- > Absetzen des Aufrufs ohne blockieren
- > Es muss später synchronisiert werden (Rückgabewert, Programmfluß)
- >

## Polling:

Objekt/Schnittstelle zum  
Nachfrage ob Ergebnis vorliegt  
→ im selben Thread

SII: AMI Polling

DII: deferred synchronous

## Callbacks:

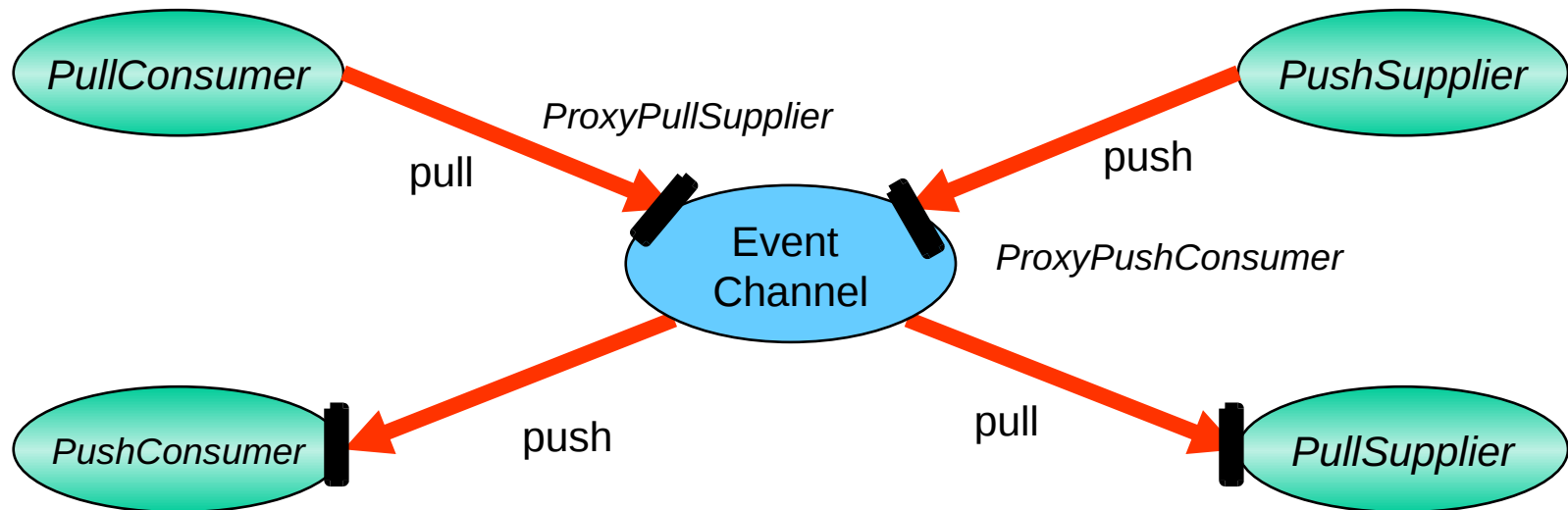
Methode, die mit dem Ergebnis  
aufgerufen wird  
→ anderer Thread

SII: AMI Callbacks

DII: manuelle Callbacks mit  
„oneway“ Funktion

# Event Channel

- > Entkopplung von Ereignisproduzenten und -verbrauchern durch zwischengeschalteten **Event Channel**
  - > Beide Verarbeitungsmodelle an „beiden“ Seiten möglich
  - > Mehrere Erzeuger und Verbraucher können sich registrieren
  - > Übertragung der Ereignisse durch normale synchrone statische CORBA-Aufrufe



# Kombinationen

		Supplier	
		Push	Pull
Consumer	Push	Information driven Event driven  asynchrone Ereignisse, die sofort weitergeleitet werden	Event channel als aktiver, intelligenter Agent
	Pull	Queue oder Puffer  Bsp.: SMTP + POP3	Demand driven mehrstufiges Client/Server System  Bsp.: Web Proxy

# Fragen?

