Scoping in Event-Based Systems

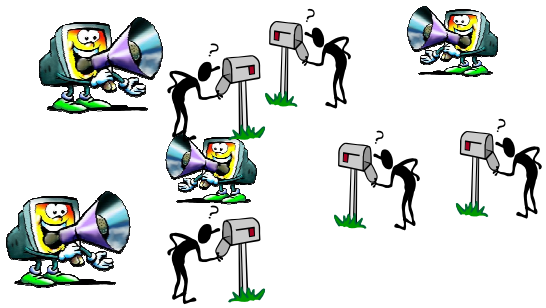
TECHNISCHE UNIVERSITÄT DARMSTADT

Ludger Fiege
Databases and Distributed Systems Group

Event-Based Communication

- Loose coupling is good/desired!
- But many inherent complexities
  - Lost control of data/control flow
  - Only implicit communication
- "Traditional" methods cannot be readily applied
  - Engineering: side effects, encapsulation
  - Management: no structure, no system parts
  - Security: no groups of trust
- Scalability is impeded by complexity

What happens?

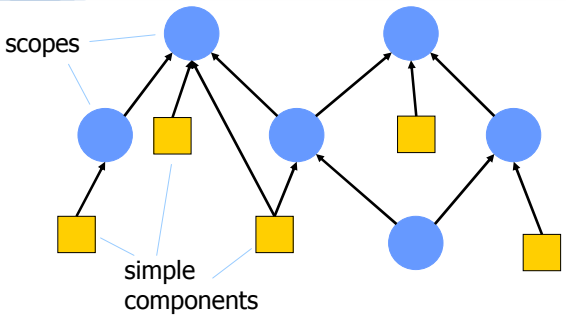


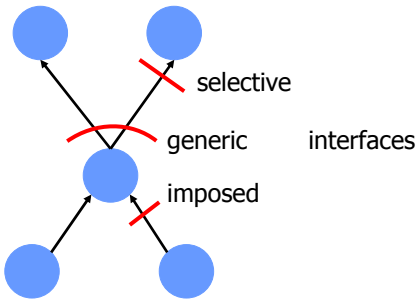
Visibility

- Control notification visibility
- … at the heart of the problems

Scoping

- Build component groups recursively
  - Directed, acyclic graphs of clients and scopes
- Scopes limit the distribution of notifications
  - Interfaces control passing between scopes
- Scopes explicitly control visibility



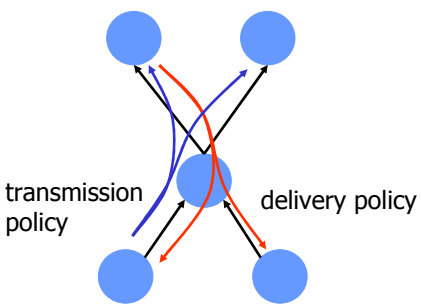


Scope Interfaces and Mappings

- Interfaces are bound to edges
  - Selective, generic, imposed interface control traffic w.r.t. specific or all superscopes, intra-scope, resp.
- Mappings transform between representations
  - Example: Serialized Java, plain text, XML, etc.
  - Generalized interfaces
  - Notification n is mapped to n, n' or ε
- Scope-specific representations of notifications
  - Heterogeneous applications
  - Efficiency improvements

Transmission Policies

- Adapt transmission in scope graphs
  - Delivery semantics tailored to applications
- Example: 1-of-n delivery, postpone delivery,…



Implementation as part of Rebeca

- Addressing scopes: extensions of issued filters
- Broker Scope: scopes are brokers
- Integrated Routing:
  - as part of the middleware
  - extend routing tables, create overlay per scope