

Broker

Context

Distributed, heterogeneous systems with independent components.

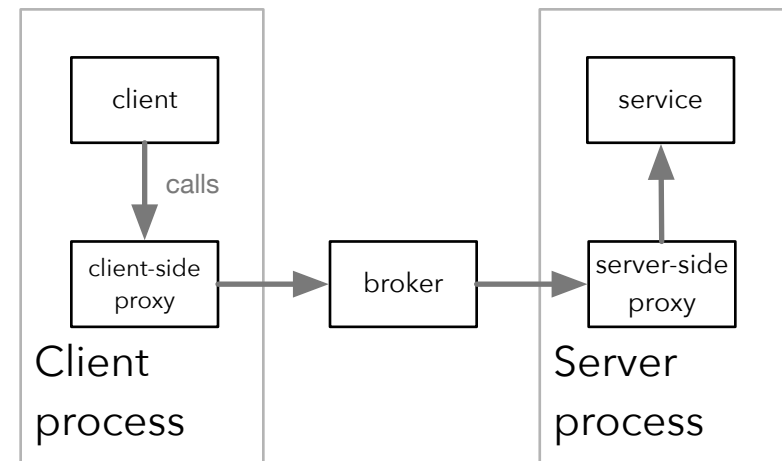
Solution

- Introduce a *broker* between *clients* and *servers*.
- *Services* register themselves with the broker.
- *Proxies* and *bridges* may help cross network types.
- Client proxy presents a function-call interface over remote calls.

See also: Peter Deutsch's "Fallacies of Distributed Computing"

Forces

- Need to communicate across process boundaries.
- Possibly heterogeneous network or transport.
- Desire for transparent remote access.
- Run-time replacement, exchange, or substitution of components.
- Architecture should hide implementation details from users.



Question: Is this an antipattern?