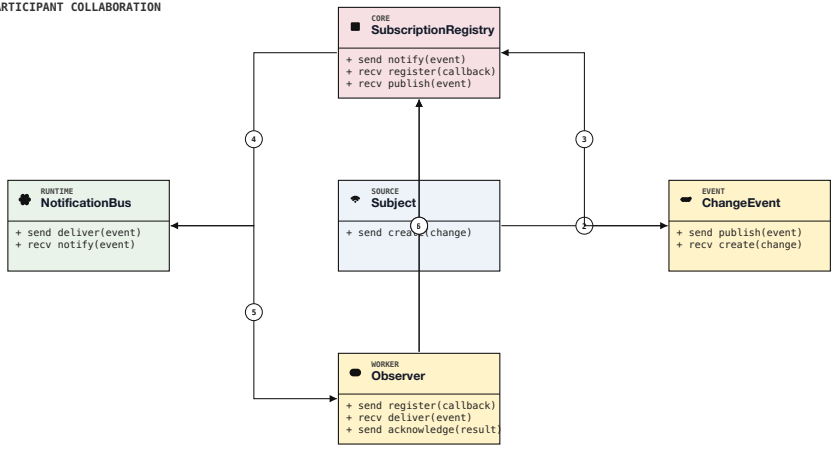
# C02 Observer

***Communication · A² A1 · broadcast role collaboration***

Notify registered dependents of changes.

**STRUCTURE · PARTICIPANT COLLABORATION**




**PATTERN INTERFACE**

***context***
A subject changes state or emits lifecycle information.

***trigger***
The subject publishes a change notification.

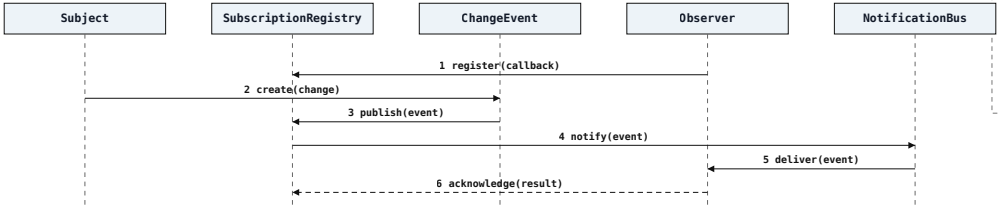
***structure***
Subscription registry, event payload, and observer callbacks.

***effect***
Currently registered observers are notified under declared ordering semantics.

**CONTRACT ANNOTATIONS**

| | |
|---|---|
| ***requires*** | Observer identity and subscription lifetime are defined. |
| ***ensures*** | Notification follows the event and unsubscribed observers no longer receive it. |
| ***invariant*** | Observer failure does not corrupt subject state. |
| ***violation*** | Callback errors and cascades are isolated or propagated by explicit policy. |

**INTERACTION**




**MESSAGE TRACE** `1 register(callback) → 2 create(change) → 3 publish(event) → 4 notify(event) → 5 deliver(event) → 6 acknowledge(result)`

Evidence: 32/41 repositories, 29/36 families; 32 trajectories (16 multi-checkpoint); exact audit: not sampled.

Variation: Event emitter, listener, hooks, and stream observers.

Negative control: Polling the subject rather than notification is not the observer pointcut.