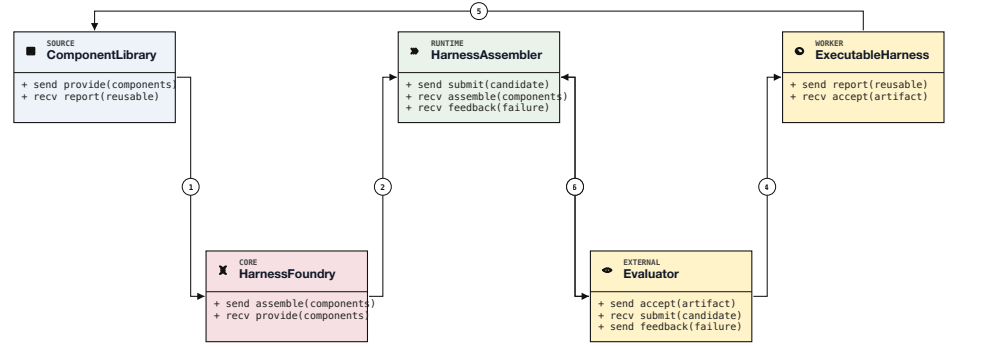
# E05 Composable harness foundry

***Evolution · A² A1/A3 · pipeline role collaboration***

Construct and evolve reusable harness components.

**STRUCTURE · PARTICIPANT COLLABORATION**




**PATTERN INTERFACE**

***context***
Harness behavior is assembled from independently typed components.

***trigger***
Components are selected, composed, or replaced.

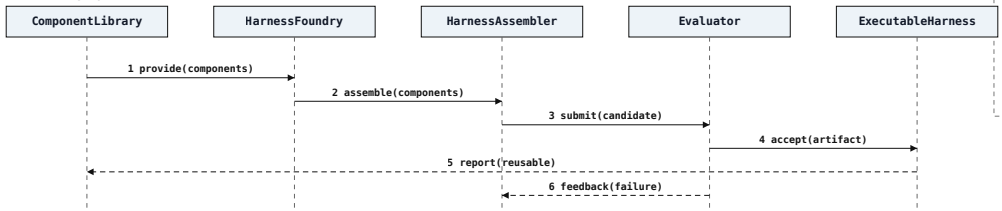
***structure***
Component interface, composition builder, evaluator, and artifact registry.

***effect***
The built harness is executable and its component provenance is recoverable.

**CONTRACT ANNOTATIONS**

| | |
|---|---|
| ***requires*** | Components have compatible contracts and composition operators. |
| ***ensures*** | Invalid composition is rejected before execution. |
| ***invariant*** | Reusing a component does not silently reuse privileged state. |
| ***violation*** | Build conflicts, incompatible versions, and evaluation failure are explicit. |

**INTERACTION**




**MESSAGE TRACE** `1 provide(components) → 2 assemble(components) → 3 submit(candidate) → 4 accept(artifact) → 5 report(reusable) → 6 feedback(failure)`

Evidence: 2/41 repositories, 2/36 families; 2 trajectories (1 multi-checkpoint); exact audit: 0 supported, 0 unresolved, 1 rejected.

Variation: Typed processors, foundry DSL, component search, and immutable builders.

Negative control: Separate helper functions without component contracts are insufficient.