

Event Storming an Architecture

Turn an agreed Process Level event flow into a committed design – aggregates, bounded contexts, crossing events, policies, and anti-corruption layers – that the implementers draw together.

DURATION

3h 15m

GROUP SIZE

—
people

WHAT YOU BRING

A Process Level event wall (original or redrawn from photos) physically in the room, plus the Process Modelling sticky palette.

WHAT YOU LEAVE WITH

- Aggregates and bounded contexts drawn on the wall
- Crossing events and commands classified per boundary
- Explicit policies and read models for each reaction
- Anti-corruption layers named for every external system

WHO TO INVITE

- **Facilitator.** Event Storming experience plus enough design background to spot aggregates that collapse under their own invariants.
- **Developers and architects.** The heart of the room – they draw the boundaries and commit to the design in front of each other.
- **Domain expert.** Vetoes clusters that won't survive contact with the business; doesn't propose boundaries.
- **Product owner.** Turns the output into backlog shape and vocabulary in the days after the session.
- **Group size 4-8.** Architecture is thinking-hardest work; above eight voices the argument space fragments and decisions stop landing.

USE WHEN

Process Level clarified a flow and you're about to build or re-architect it

Splitting a monolith and need to agree on the seams

A new service must decide what it owns, subscribes to, and emits

Two teams own overlapping responsibilities and need boundaries before release

AVOID WHEN

The underlying flow isn't clear – run Process Level first

The domain is unfamiliar to half the room

No developers with design responsibility are present

The team lines are fixed outside the room and can't be challenged

How the session runs

● Phase 1 – Orient to the wall (15 min)

Walk the Process Level wall end-to-end, accept small corrections, then close the door: the events are ground truth and today's question is where the code boundaries go.

● Phase 2 – Identify aggregate candidates (35 min)

Teach five tests (crash, shared-rule, same-ID, Conway's-Law, long-running-process) and let small groups place coloured dots to cluster events into candidate aggregates before any line is drawn.

- **Phase 3 – Draw boundaries (30 min)**

Thick-marker each agreed cluster on the paper, deciding per boundary whether it's one aggregate or a bounded context of a few. State the no-event-causes-event and no-command-causes-command rules out loud.

- **Phase 4 – Crossings: commands vs events (25 min)**

Walk every arrow that crosses a boundary and classify it as a crossing event or a crossing command. Tilt the ratio toward events and reshape commands into publish-subscribe where possible.

- **Phase 5 – Make policies and read models explicit (30 min)**

For every crossing event, add a purple 'when X, then Y' policy inside the subscribing boundary, plus a pale-green read model wherever the policy needs data to decide. Flag long-running policies.

- **Phase 6 – External vs internal systems (15 min)**

Walk the yellow actor notes and mark each as owned or integrated. Ring the two groups in different colours and name an anti-corruption layer explicitly on the wall for every external system.