THE WORKSHOP

C4 Modelling

Produce a small, nested set of Context, Container, and selective Component diagrams that tell one consistent story about a system at increasing levels of detail, using notation the whole team can maintain.

DURATION

2h 40m

GROUP SIZE

—

people

WHAT YOU BRING

A whiteboard and markers; ideally an Event Storming wall or equivalent shared mental model of the system to synthesise from.

WHAT YOU LEAVE WITH

- A System Context (C1) diagram naming users and external systems
- A Container (C2) diagram showing what runs and where state lives
- Selective Component (C3) diagrams for containers that earn one
- A chosen tool, named owner, and review cadence for the durable version

WHO TO INVITE

- **Facilitator.** Holds the zoom level, parks off-level detail, and keeps the group on one diagram at a time.
- **Architects / tech leads.** Externalise the architectural model they carry and surface disagreements they didn't know existed.
- **Developers on the system.** At least two. The people who write the code own the diagram, or the team will ignore it.
- **Operations representative.** SRE or platform engineer. Essential for any deployment-view work and naming real infrastructure.
- **Technical writer (optional).** Keeps the diagram alive afterwards and translates whiteboard shorthand into the durable artefact.

USE WHEN

An Event Storming an Architecture session needs turning into durable documentation

Onboarding costs rise because the architecture only lives in people's heads

Compliance or security have asked for a real architecture diagram

Two teams argue about architecture at different zoom levels without realising it

AVOID WHEN

The team hasn't decided what the system does — run Event Storming first

You want a detailed infrastructure inventory or a single-class code view

Scope is one script, one Lambda, or a single-file utility

One architect will draw it alone and email it round for approval

How the session runs

Phase 1 — Scope and level framing (15 min)

Agree which diagrams this session will produce and why, based on the questions the team actually has. Check for an Event Storming wall and decide whether this is synthesis or discovery.

Phase 2 — Draw the System Context (C1) — 25 min

Draw the one-box system with its users and external systems around it, labelling every arrow. No internal structure, no databases — the diagram an outsider can read at a glance.

Phase 3 — Draw the Container diagram (C2) — 45 min

Open up the box and draw every deployable unit and datastore with the arrows between them. Use the Event Storming wall's bounded contexts as the reference for container boundaries.

Phase 4 — Zoom into Components (C3) — selective, 30 min

Walk the containers and decide which earn a component diagram, vetoing the rest. Draw C3 only for containers whose internal shape a developer cannot pick up from the code alone.

Phase 5 — Review and name-check — 20 min

Pressure-test the diagrams against production reality, missing scheduled jobs, and the Event Storming wall. Mark any drift or follow-ups in a different colour on the whiteboard.

Phase 6 — Wrap-up, tooling, owners — 15 min

Commit to a single tool, a named owner, a deadline for the durable version, and a maintenance cadence. Do not leave the room without each of these decided.