That's fine, until the next replay. If you design it right, email is mandatory... it shouldn't be optional. So you System relies on the email to be available. In case of a replay though, we start with the older Events.. all without email. If you didn´t account for that, you´ll have a Bug in Code (even though all your tests are green, I suppose)

Suddenly every projection has to parse both the old version (no email) and the new version (with email). Otherwise you get exceptions: "email not available."

The "quick & dirty" fix? Add null checks: if email != null.

But here's the real problem - this knowledge gets lost over time. Six months later, someone looks at the code, sees email is mandatory, thinks "why the null check?" and removes it. We are back at the start.

Everything works fine. Until the next replay destroys everything (again)

Option 1: Don´t change anything - just add another Event

Instead of changing the existing Event "Item added", which should be an immutable Fact anyway, we could just add another Event fired by the same command like "Email attached to Item".

This immediately solves all versioning problems, as there is no versioning. Every consumer can now decide if the new Event should be considered or not. This is by far the simplest solution - but brings the danger of polluting the Event Stream with "Property-Sourced" Events.. if we add the Phone Number, we´ll have "Attached PhoneNumber to Item" and so on.

The Pattern you see here is called the "left-chair" and one of my [4 Anti-Patterns](#).

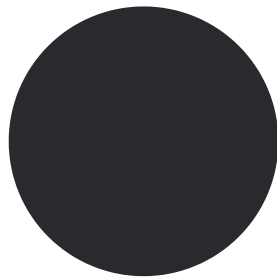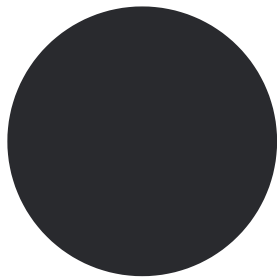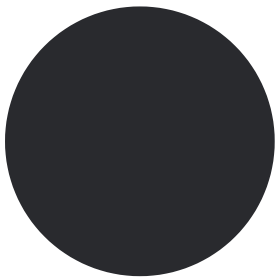If the new Event is meaningful, also to a business person, I´ll always choose this option.

Add Item

Einladung von Gaëtan B. angenommen. ✕

Einladung von Gaëtan B. angenommen.
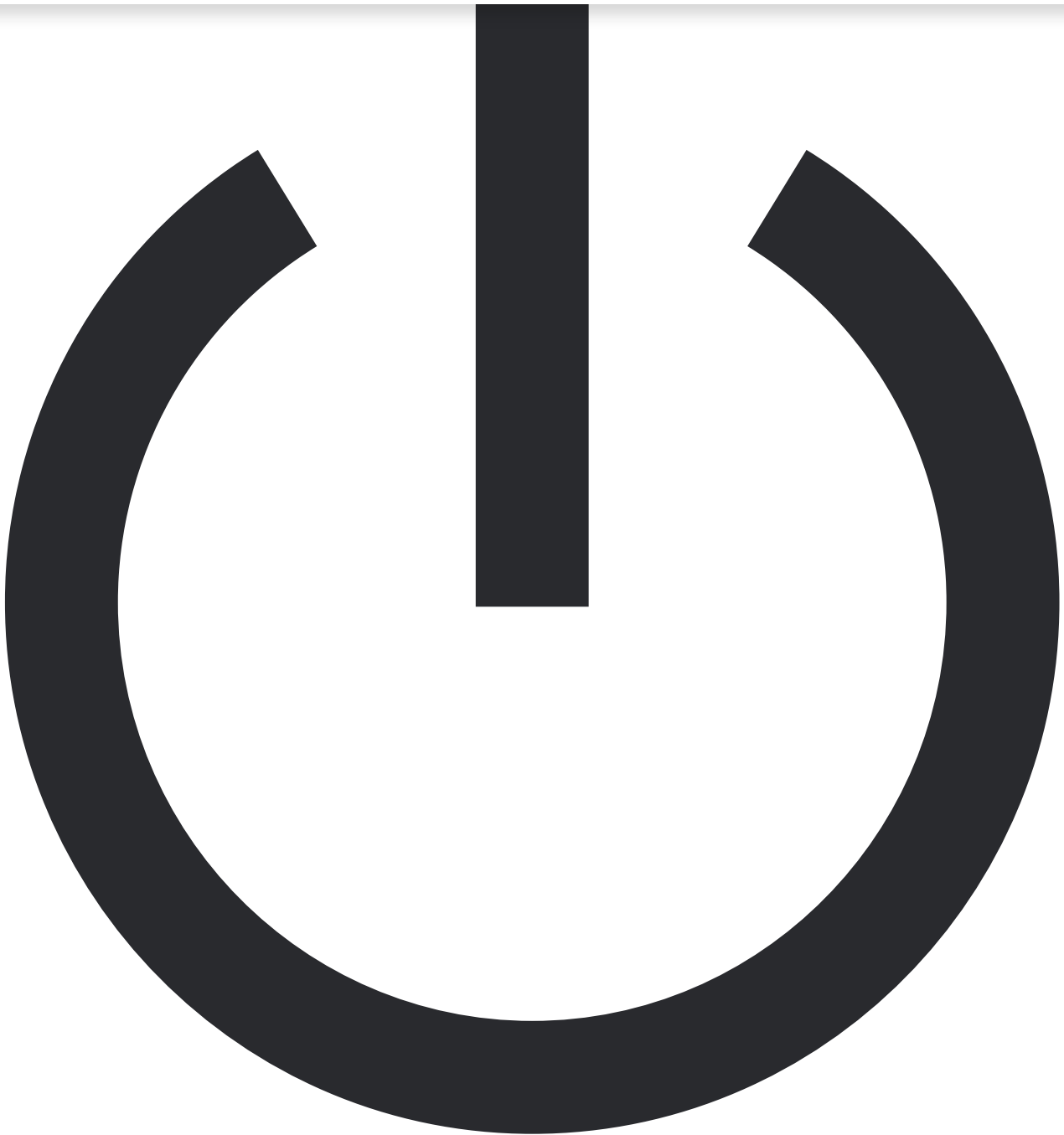
Record screen only

✅ Einladung von Gaëtan B. angenommen.

Record camera only

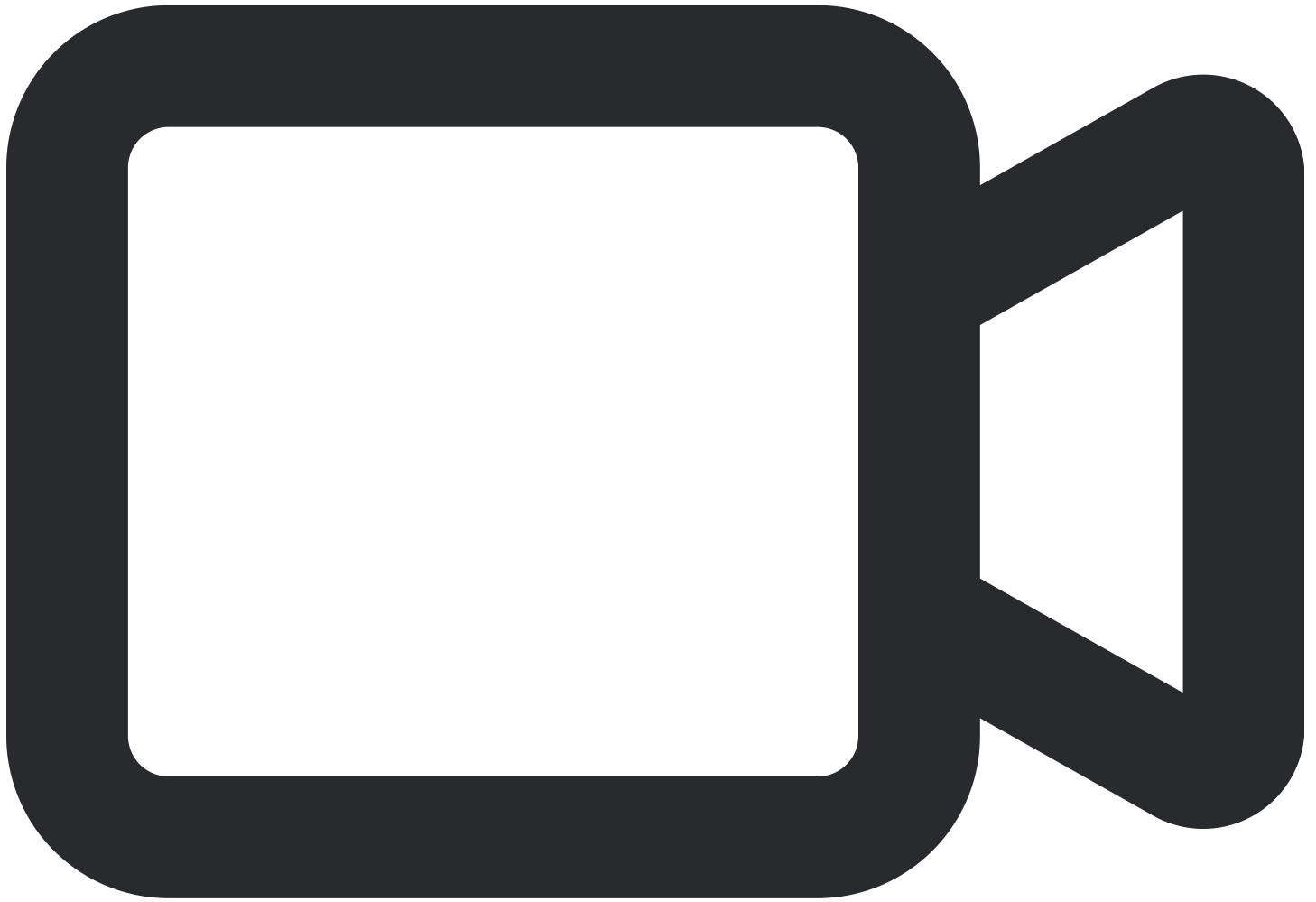✅ Einladung von Gaëtan B. angenommen.

Turn off for this site

Einladung von Gaëtan B. angenommen.

✅ Einladung von Gaëtan B. angenommen.

Record a Loom

✅ Einladung von Gaëtan B. angenommen.