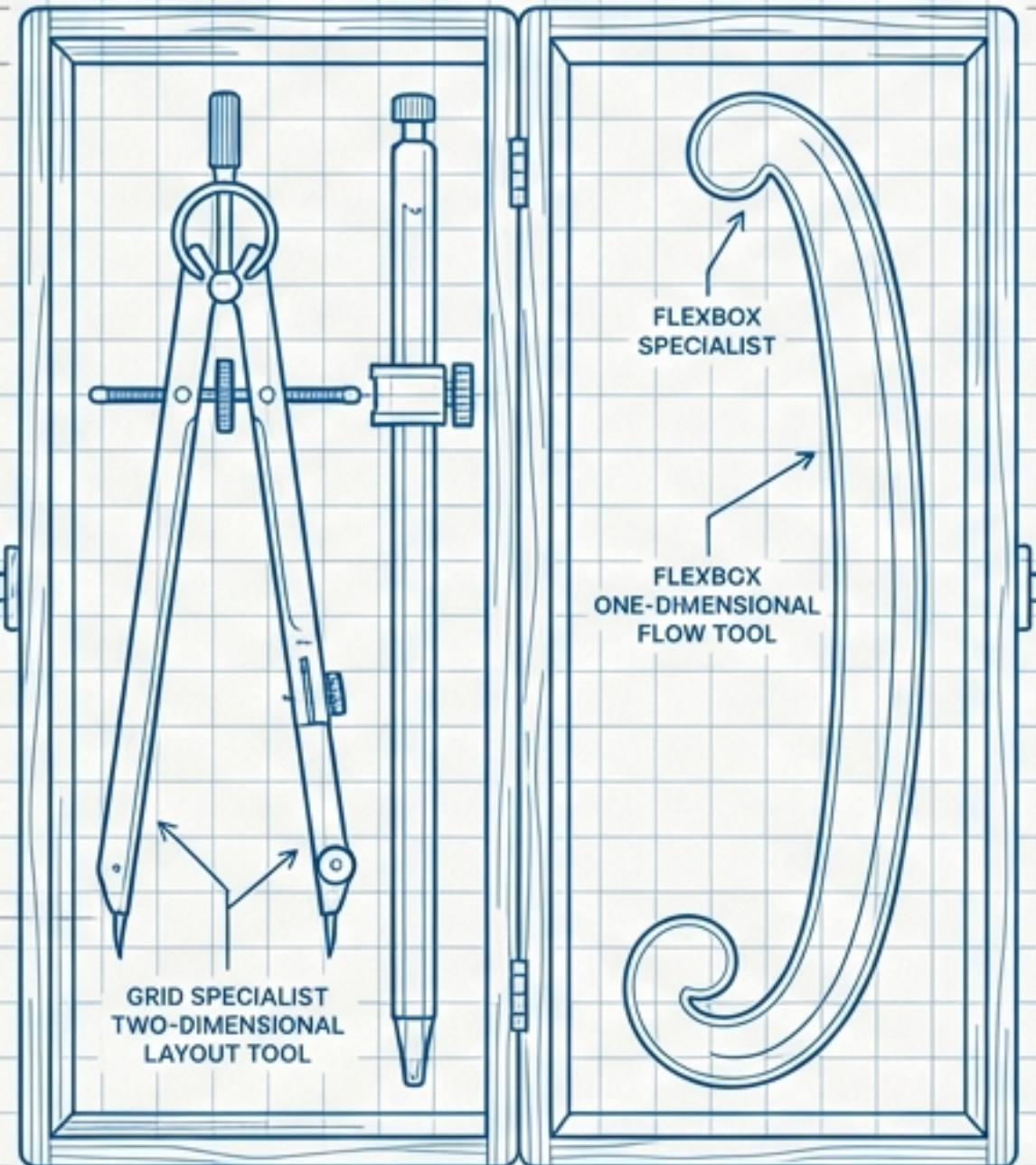


Grid vs Flexbox

Choosing the Right Tool

The Architect's Battle is a Myth.

- Flexbox and Grid are not rivals. They're specialists.
- One is for one-dimensional flow (a row or column).
- The other is for two-dimensional layout (rows and columns).



DRAWING TITLE:

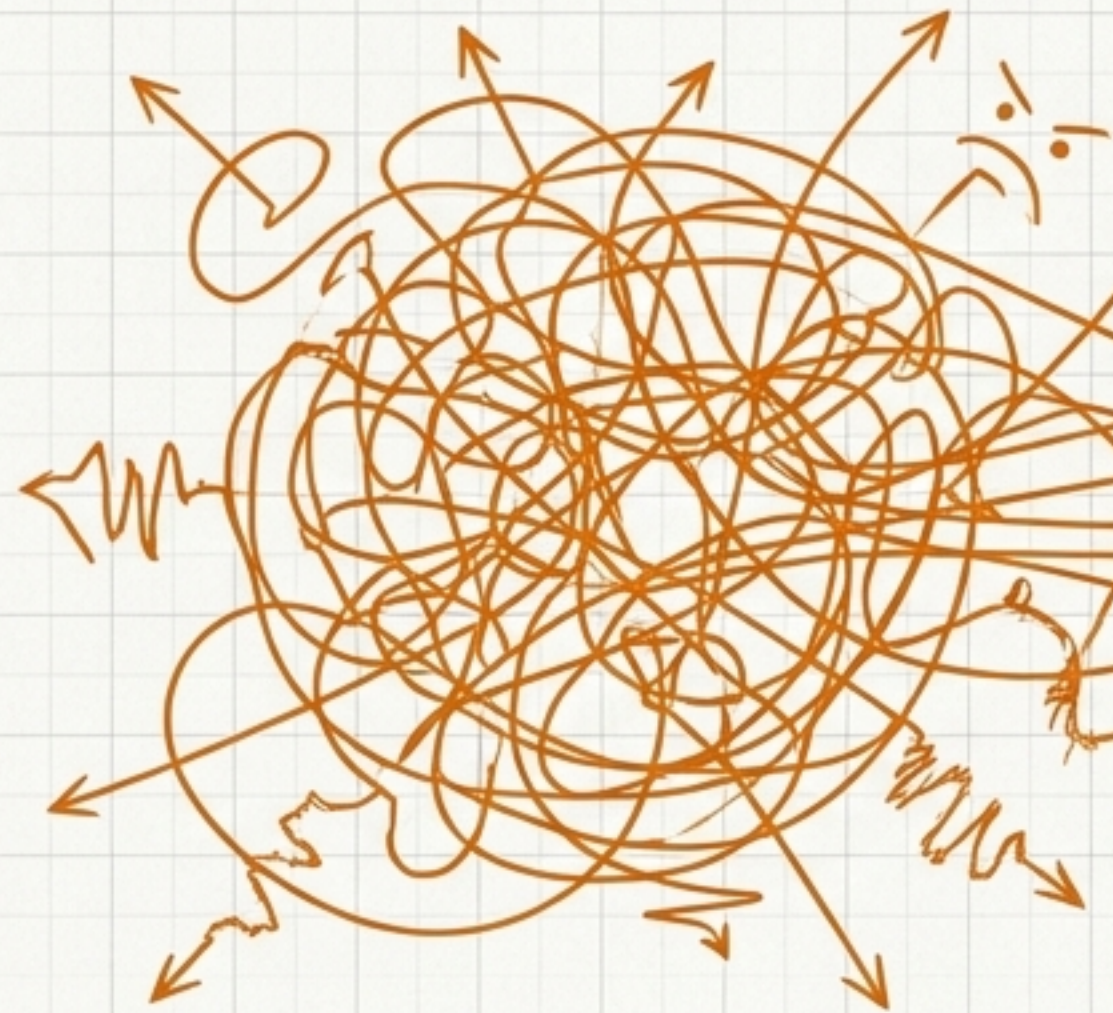
TOOLKIT SYNERGY DIAGRAM

SCALE: NTS

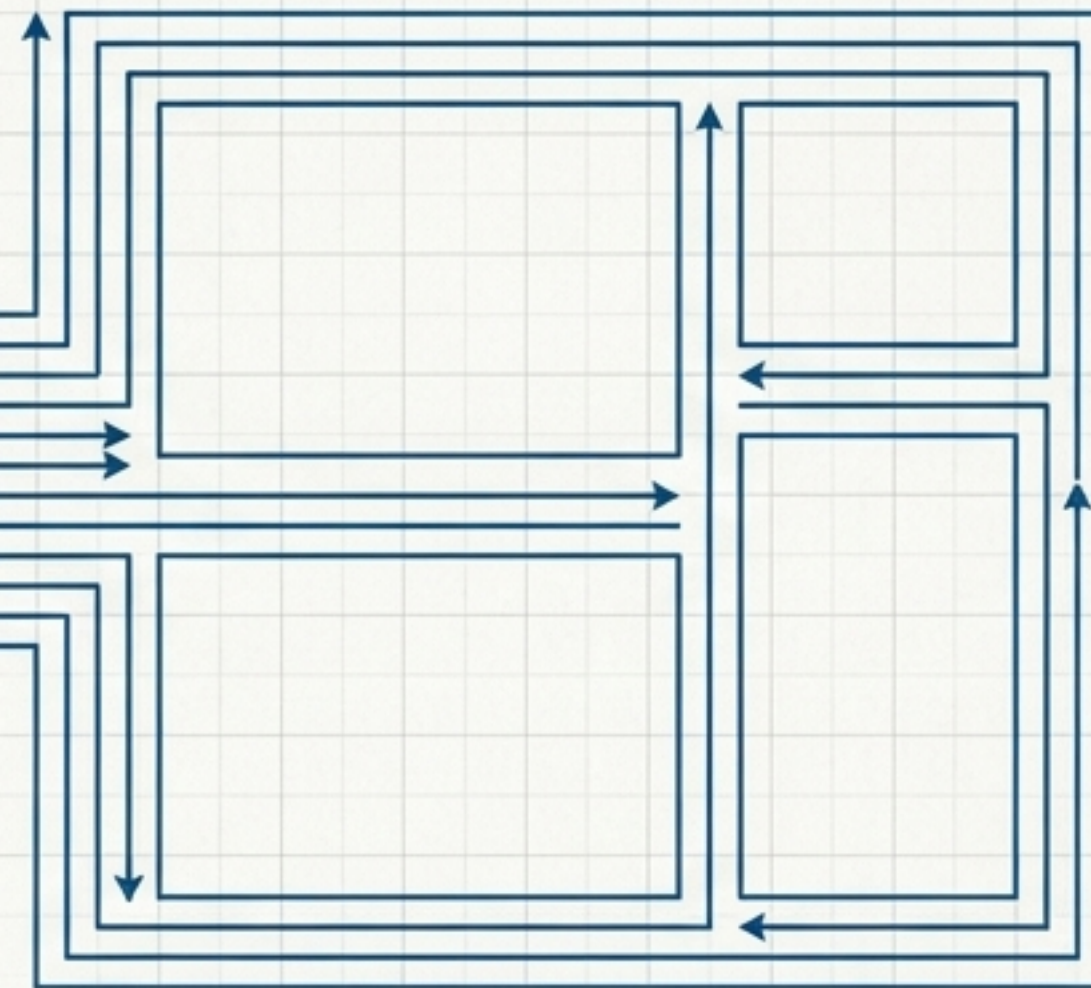
DATE: OCT 26, 2023

The Cost of the Wrong Tool

If you pick the wrong one, you start wrestling the browser... and losing.



If you pick the right tool, your CSS gets smaller.



DRAWING TITLE:

TOOLKIT SYNERGY DIAGRAM

SCALE: NTS

DATE: OCT 28, 2023

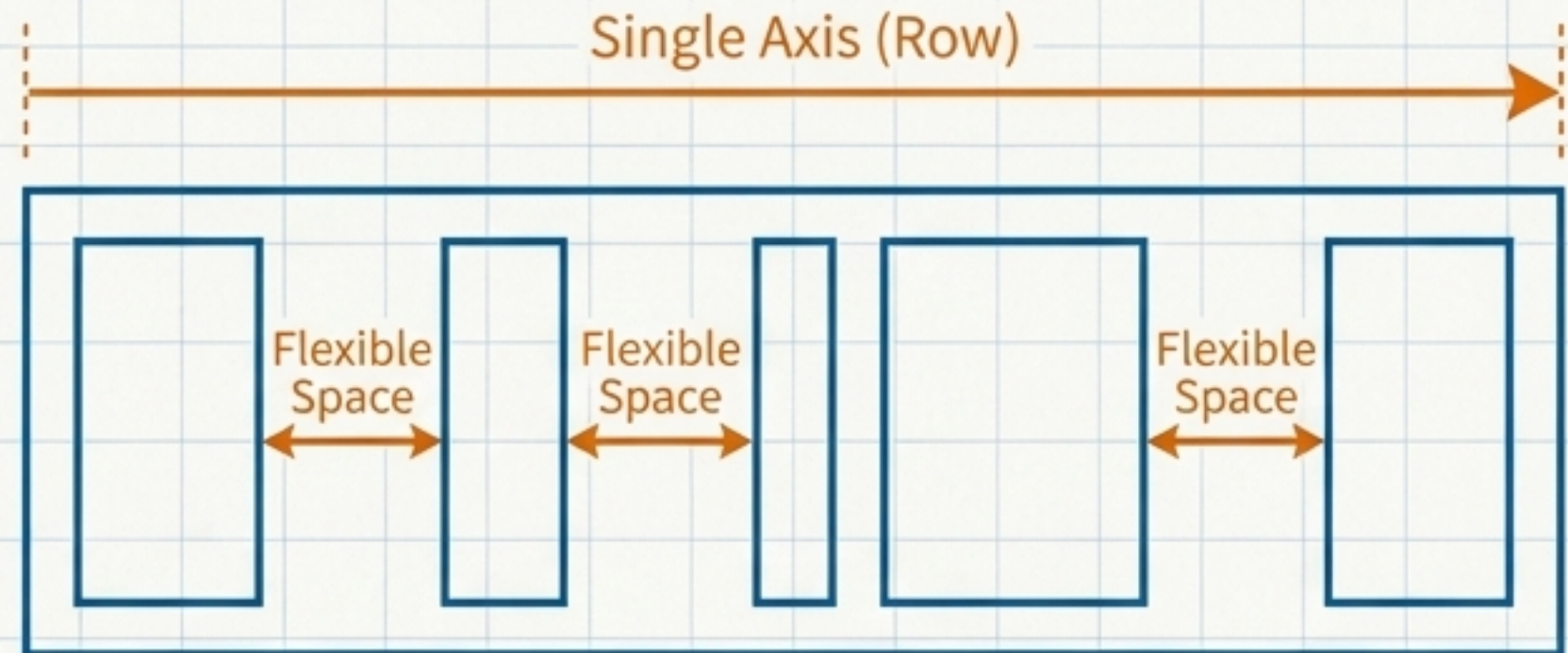
© 2023 Architectural Layout Systems. All rights reserved

NotebookLM

When to Use Flexbox

Use Flexbox when...

1. **You're laying out items along one axis.** (Either a row or a column).
2. **You want content-driven sizing.** (Items flex and grow based on the content inside them).
3. **You're building UI components.** (Aligning and spacing controls).



DRAWING TITLE: **TOOLKIT SYNERGY DIAGRAM**

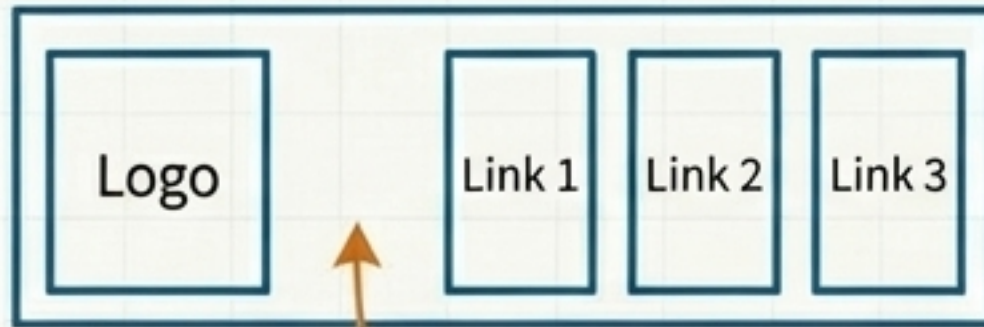
SCALE: NTS

DATE: OCT 28, 2023

© 2023 Architectural Layout Systems. All rights reserved.

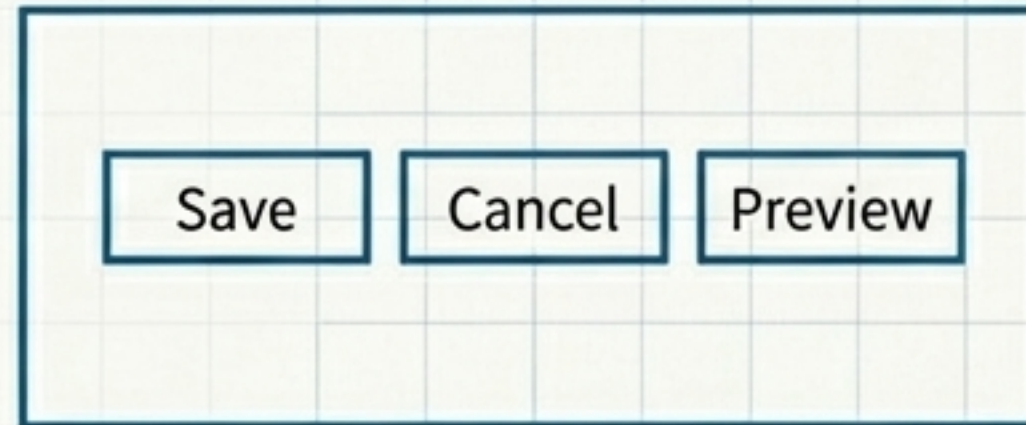
Flexbox in Action: UI Components

Nav Bars

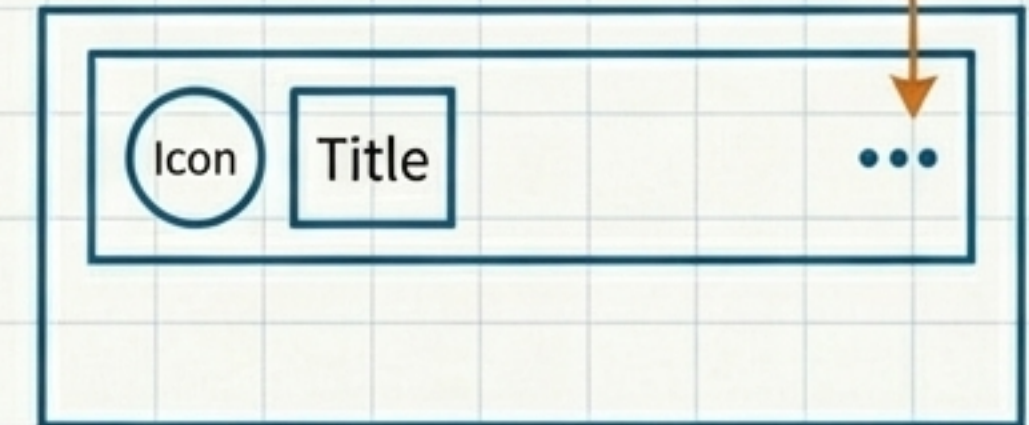


space-between
alignment

Button Rows



Card Headers



Think alignment, spacing, and flow for a single row or column.

DRAWING TITLE: **FLEXBOX EXAMPLES**

SCALE: NTS

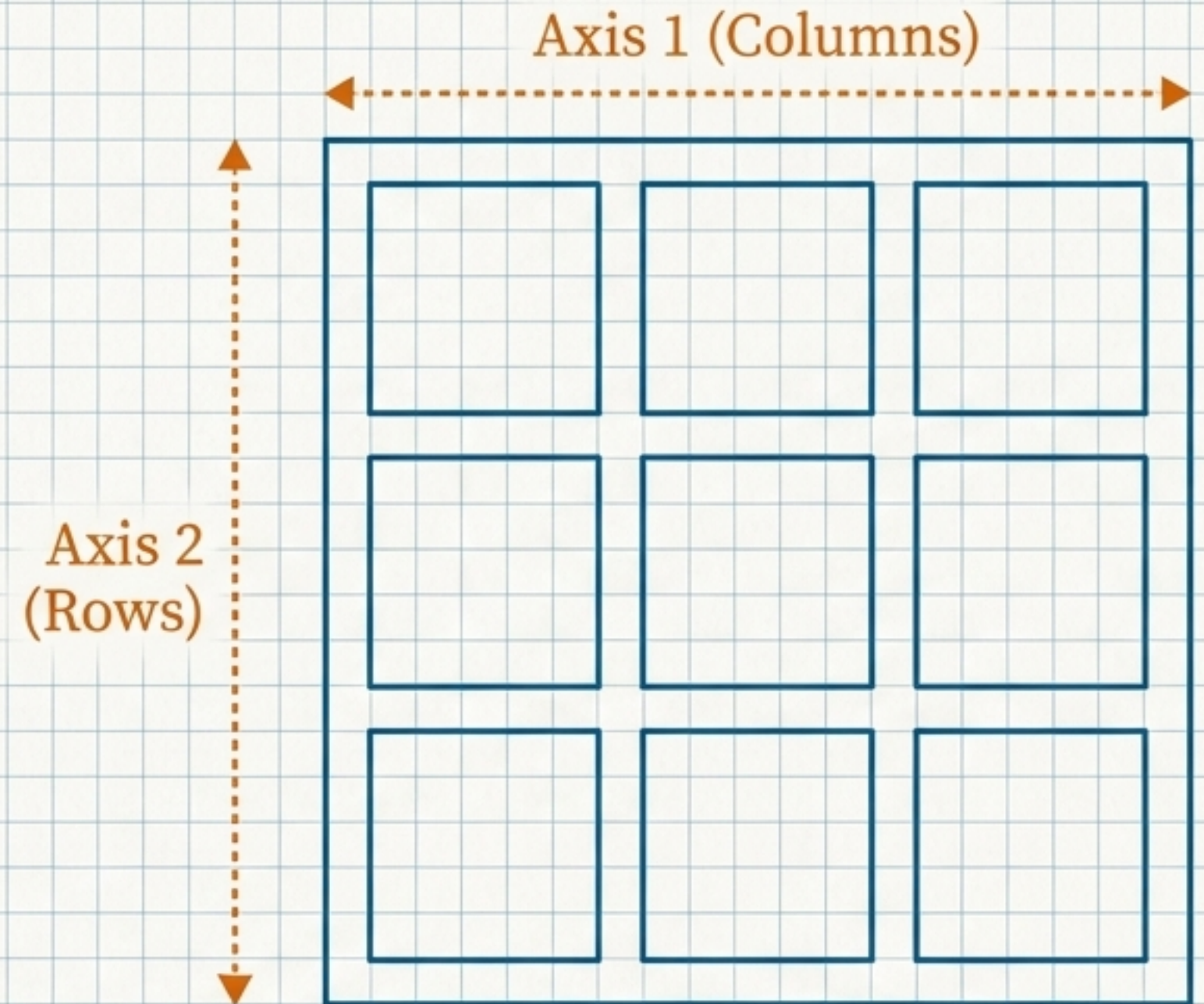
DATE: OCT 28, 2023

© 2023 Architectural Layout Systems. All rights reserved.

When to Use Grid

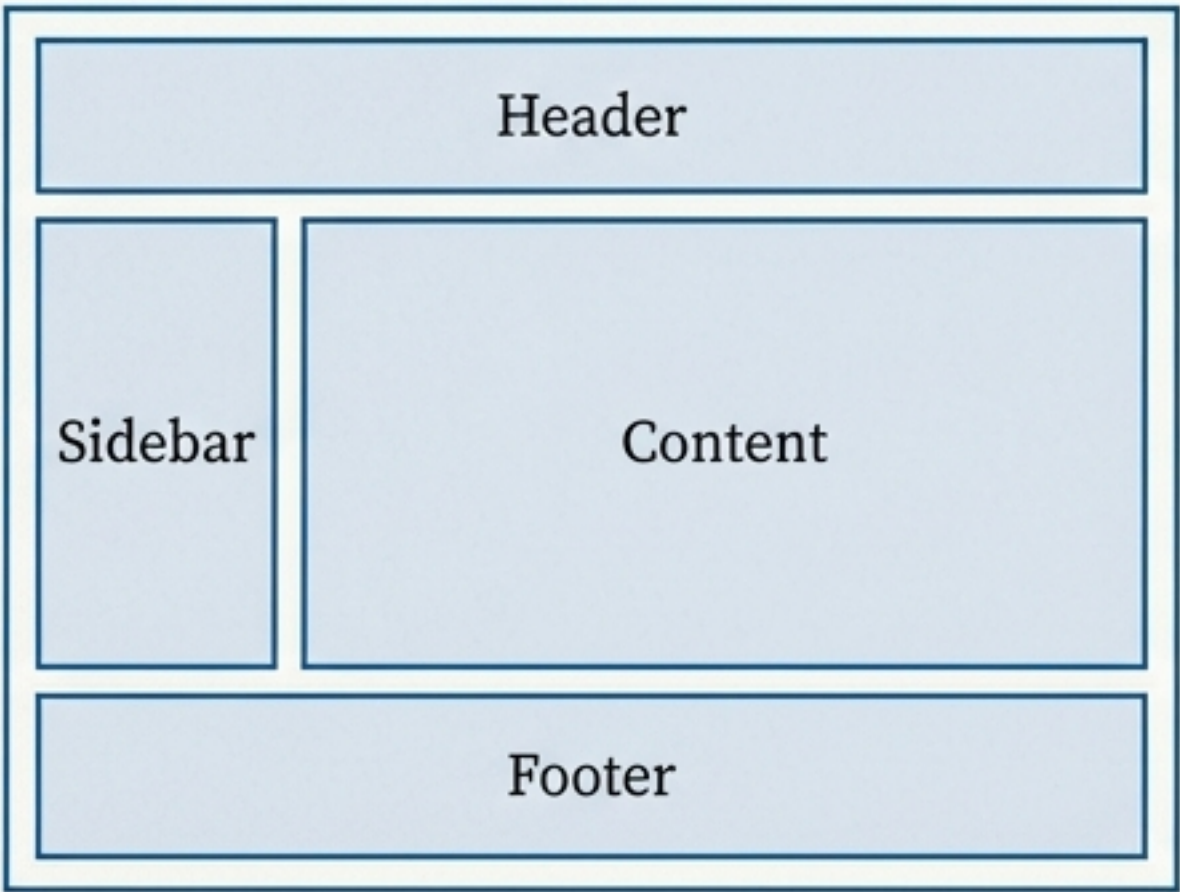
Use Grid when...

1. You need **both axes**. (You need control over both rows and columns simultaneously).
2. You want **explicit structure**. (You are defining the blueprint for a layout).
3. You want **responsive layouts that can rewire**. (Change the entire structure at different breakpoints).



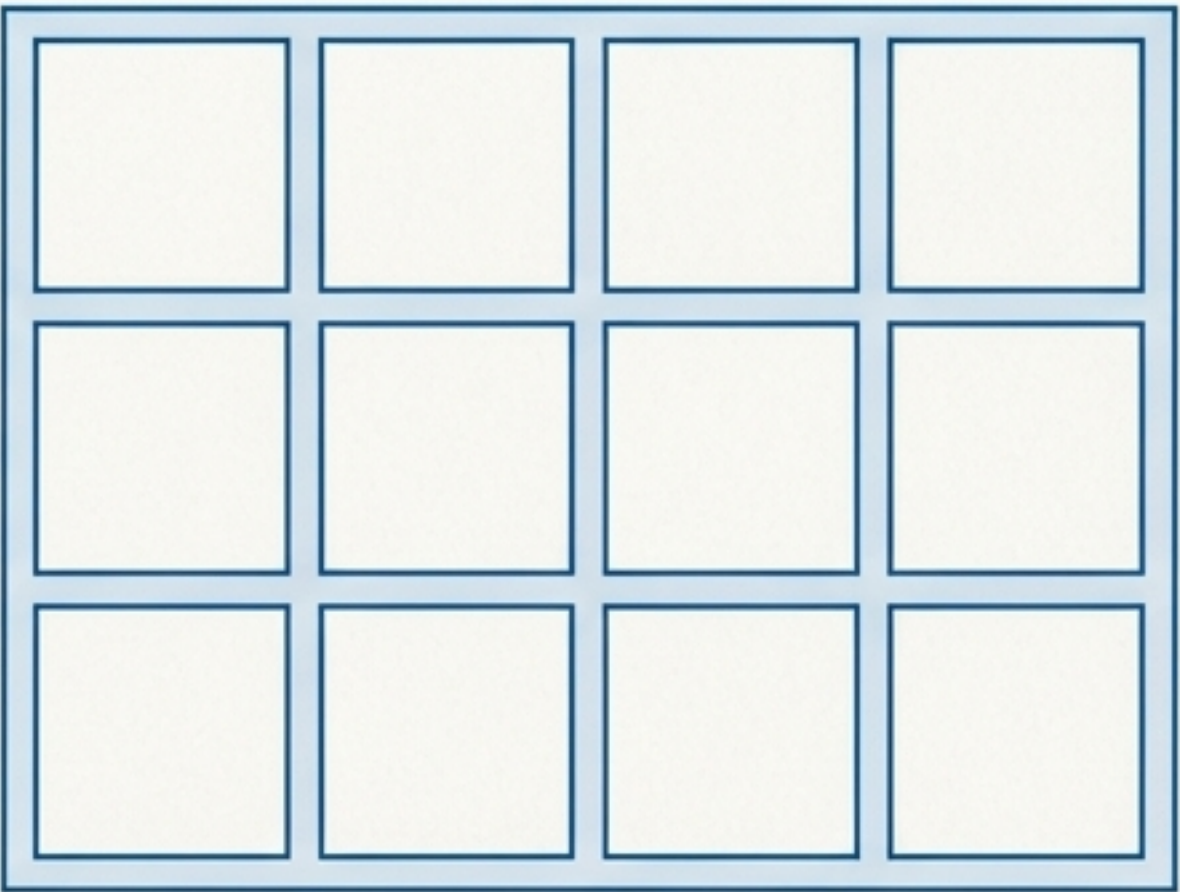
Grid in Action: Page Skeletons

Page Layouts



Page Layouts

Galleries & Dashboards

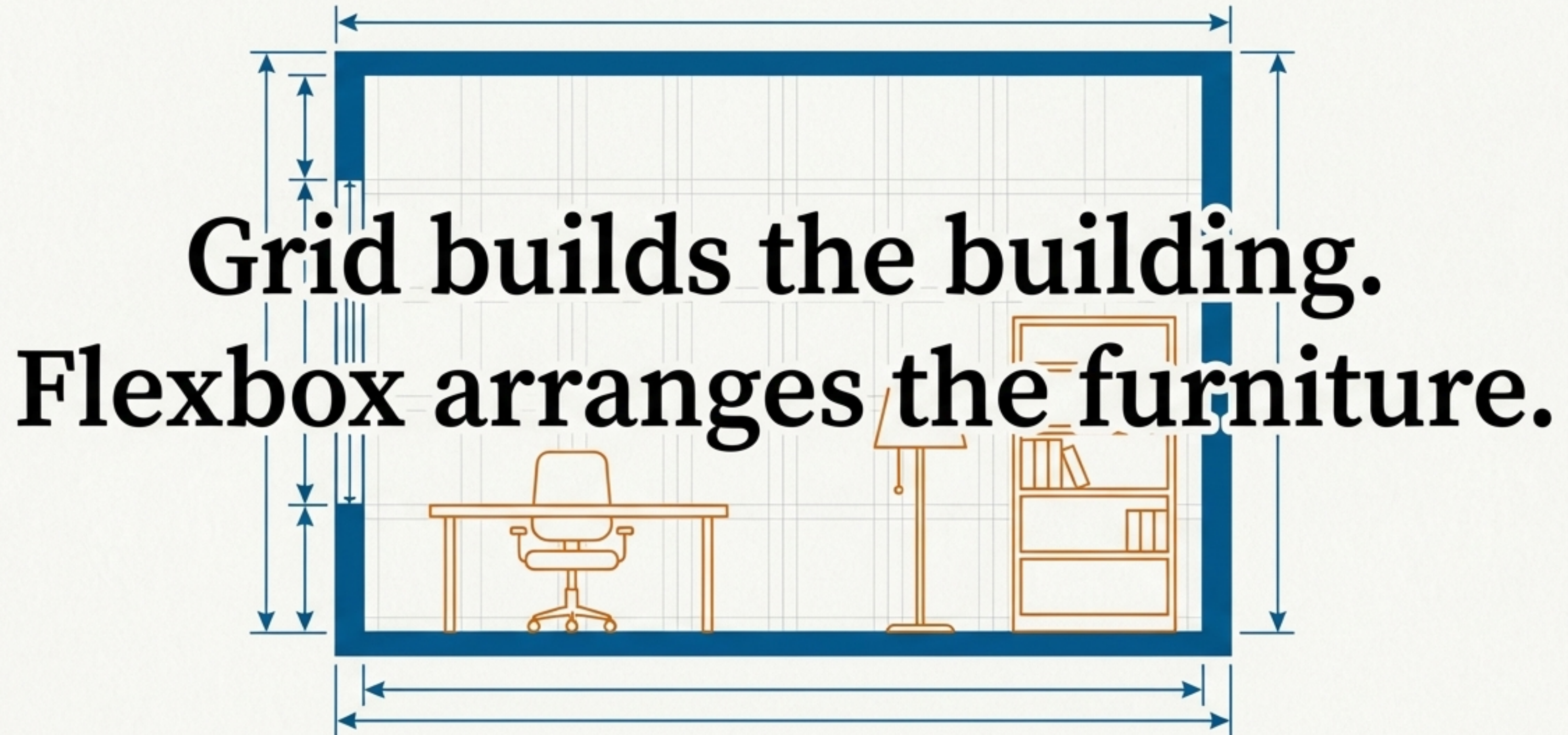


Galleries & Dashboards

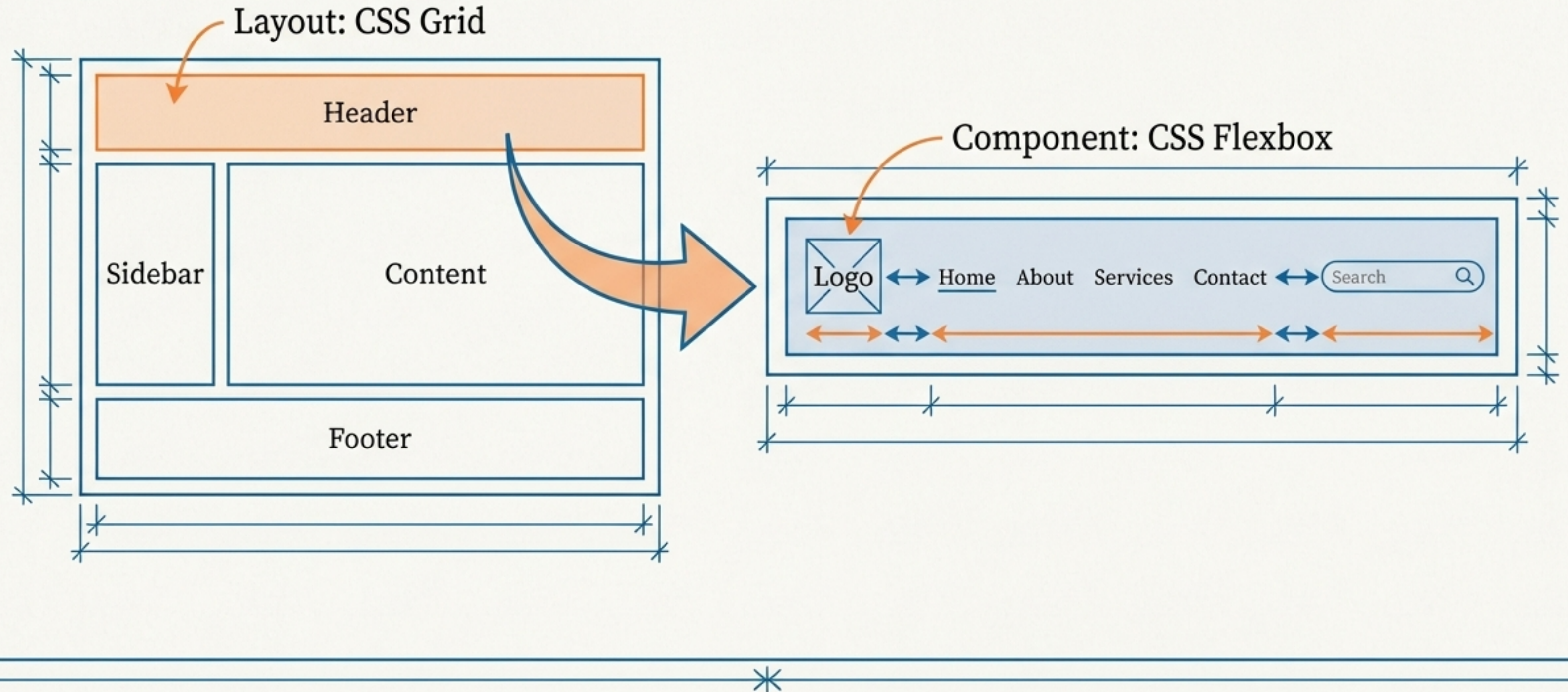
Think blueprints, regions, and overall page structure.

The Pro Move: Use Both

The most common and powerful pattern is to combine them.



From Blueprint to Component



The Architect's Decision Console

The diagram shows a table with three columns: Scenario, Call, and Why. A blue double-headed arrow spans the width of the 'Why' column. A blue double-headed arrow on the left indicates the height of the first two rows. An orange double-headed arrow on the right indicates the height of the last two rows. An orange double-headed arrow at the bottom indicates the width of the 'Call' column. An orange line with an arrow points from the 'Call' header to the 'Flexbox' entry in the first row.

Scenario	Call	Why
Navbar / Button Row	Flexbox	One axis (row). Content-driven sizing. Easy alignment & spacing.
Photo Gallery / Tile Grid	Grid	Cards arranged in rows and columns with consistent gaps.
Page Layout: Header / Sidebar / Content	Grid	Explicit regions with a readable, two-dimensional blueprint.
Dashboard: Page grid + widgets inside	Grid (for page) + Flexbox (for widget internals)	Big layout plus component-level alignment inside each widget.

The Takeaway

You don't "graduate" from Flexbox to Grid. You build a toolkit.



Choose intentionally.



Combine confidently.

Collect tools.
Build *systems*.



p.s., keep learning!