

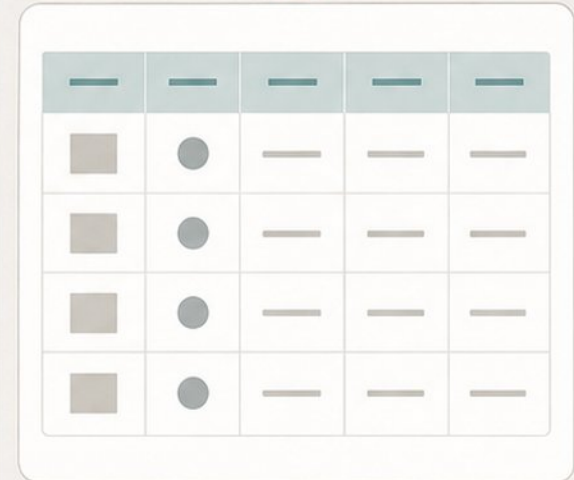
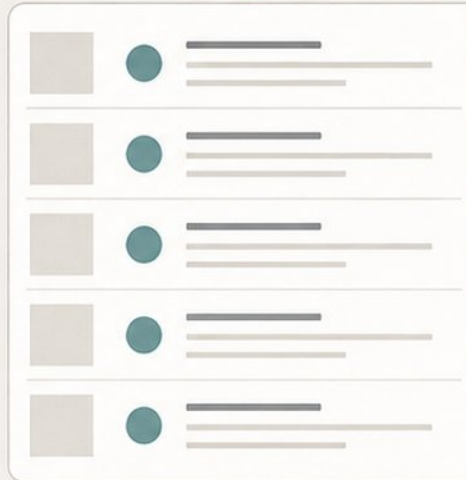
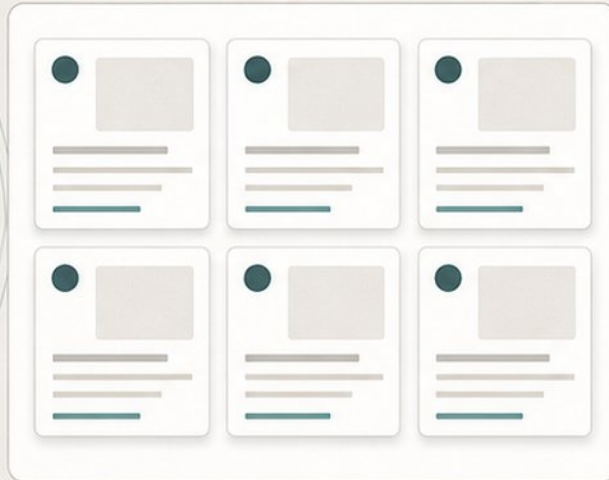
Designing Content Collections with Cards

- Define content collections: repeatable, comparable items grouped for browsing
- Cards are the dominant UI pattern for self-contained, scannable units
- Preview the journey: discovery goals → layout → interaction → governance
- Real-world examples: streaming libraries, e-commerce grids, dashboards, search results
- Learning objectives: card anatomy, layouts, comparison, accessibility, and scalable governance



Why Cards? Discovery, Comparison, and When Not to Use Them

- Cards are entry points for visual browsing — each item is self-contained
- Comparison across cards is weaker than tables; best for manageable datasets
- Choose cards for browsing, lists for linear scanning, tables for dense comparison
- Anti-pattern: forcing cards onto search results, data tables, or settings pages



The Psychology of Scanning and Comparison



- F-shaped & Z-shaped patterns guide card layout; interrupt them carefully



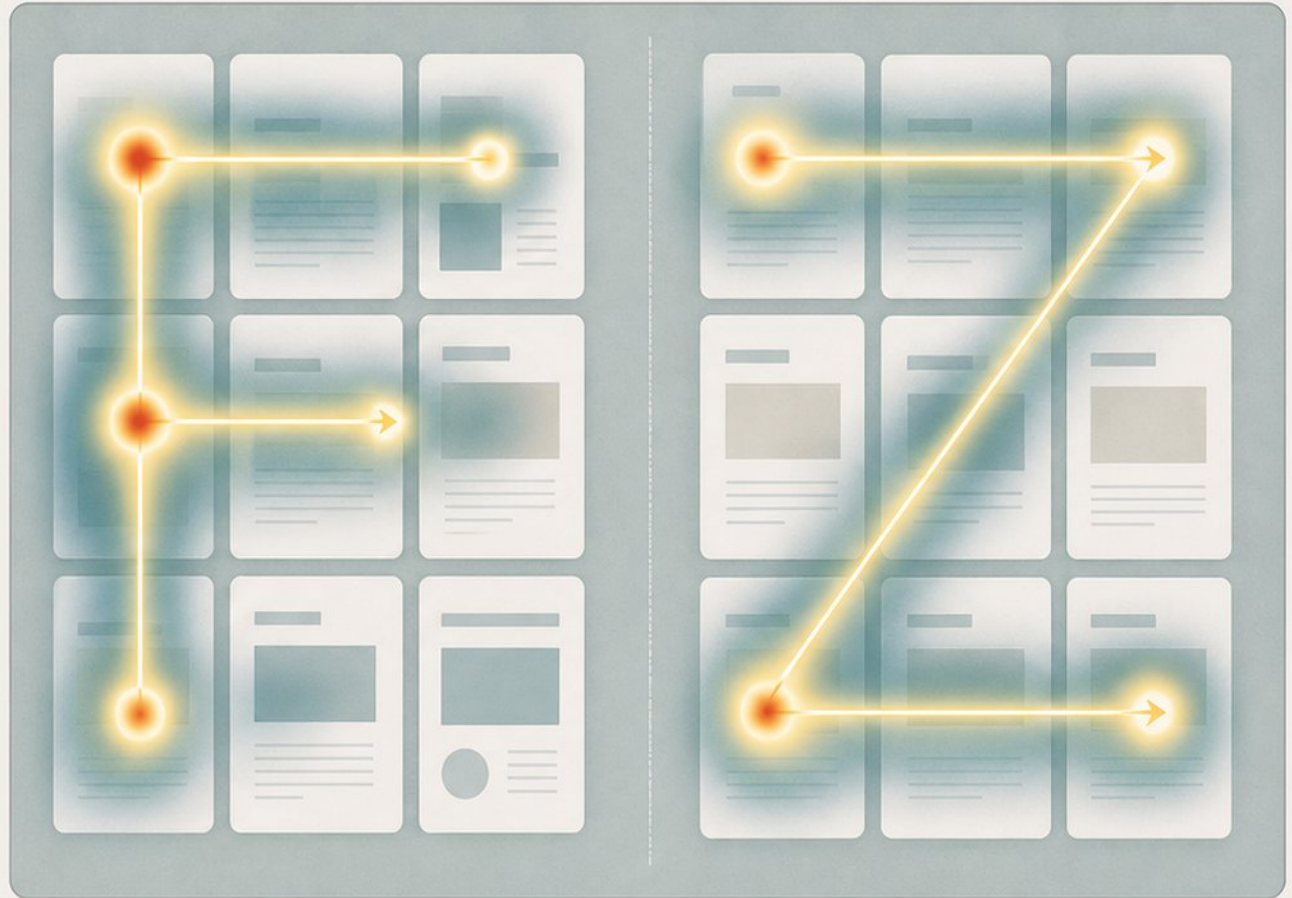
- Vertical cards suit visual browsing; horizontal cards support text-heavy evaluation



- Hick's Law, Gestalt proximity, and serial position effect shape comparison

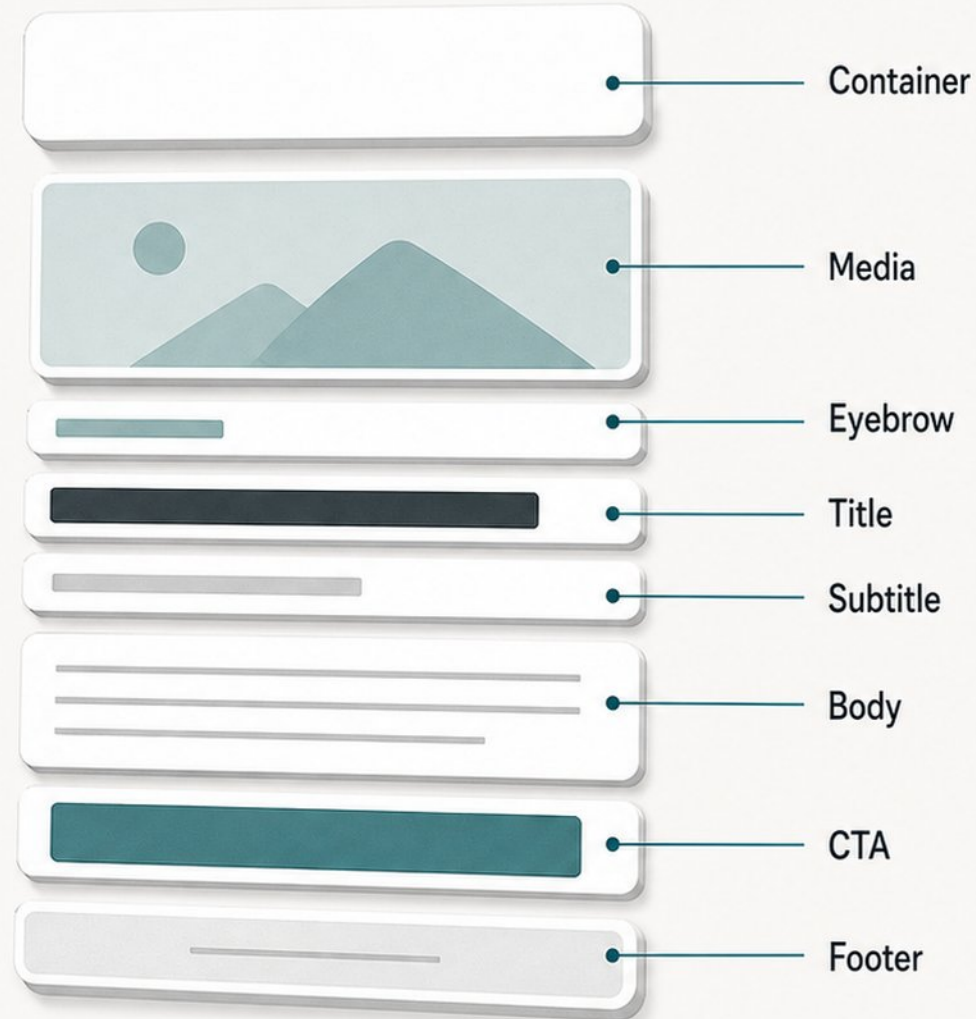


- Cards act as islands; use alignment, rhythm, and proximity to preserve flow



Card Anatomy and Content Modeling

- Atomic elements: container, media, eyebrow, title, subtitle, body, primary action, footer
- One card, one concept — decide what belongs at collection vs. detail level
- Limit summary text to 2–3 lines (under 100 characters); answer one question
- Boolean slots (media, badge, footer) extend a base card; rigid templates preserve scannability



Collection Architecture and Layout Patterns

- Grid for uniform browsing, list for linear scanning, table for dense comparison
- CSS auto-fill: columns adapt fluidly without breakpoints (minmax)
- Bento grids: cells sized by content importance, not uniform repetition
- Subgrid aligns card internals across rows; container queries enable adaptive cards
- Consistent spacing, equal-height rows, and intentional masonry for variable heights



Grid



List



Masonry



Bento



Carousel



Subgrid

Visual Attributes That Enable Comparison



Use imagery, ratios, and typographic scale to create visual hierarchy within cards



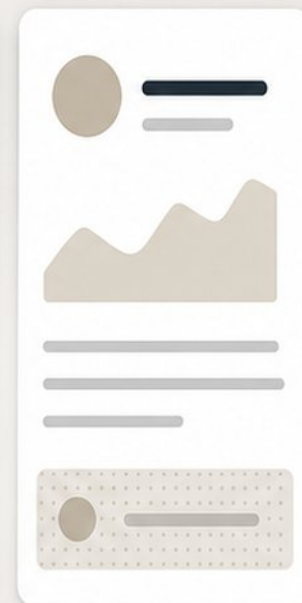
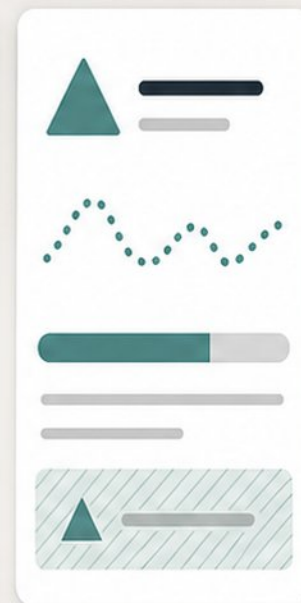
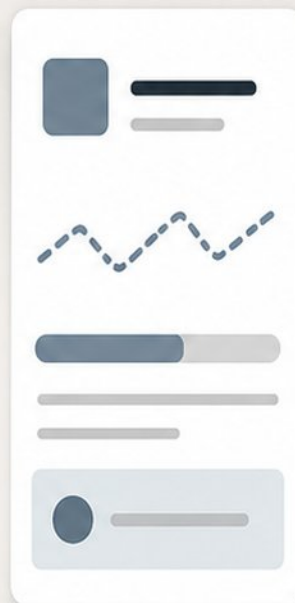
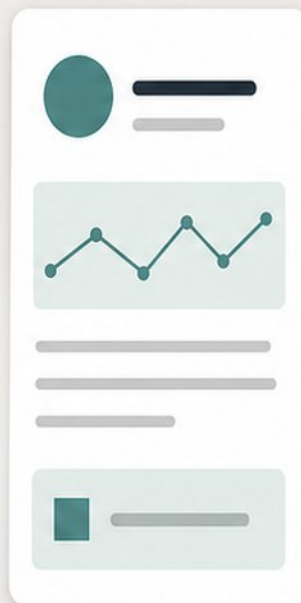
Encode meaning with micro-charts, sparklines, and badges — not visual noise



Never rely on color alone; pair with shape, label, pattern, or position



Meet WCAG 2.1 AA: 4.5:1 text contrast, 3:1 for graphical objects and UI components



Interaction Design: States, Affordances, and Filtering



- Define 6 essential interactive states: default, hover, focus, active, disabled, loading



- Hover states signal interactivity without causing layout shift



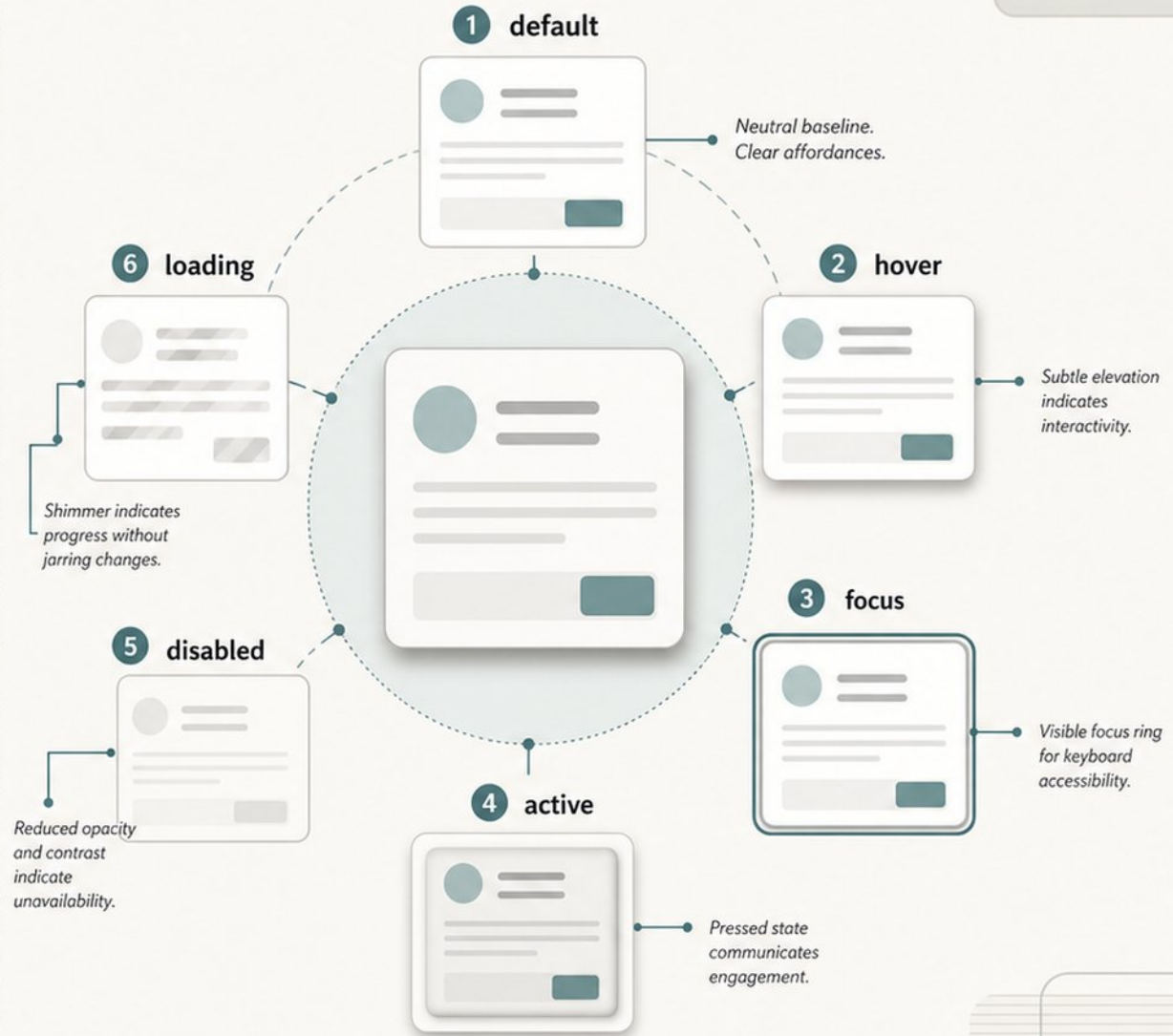
- Focus states are mandatory for keyboard users; use :focus-visible



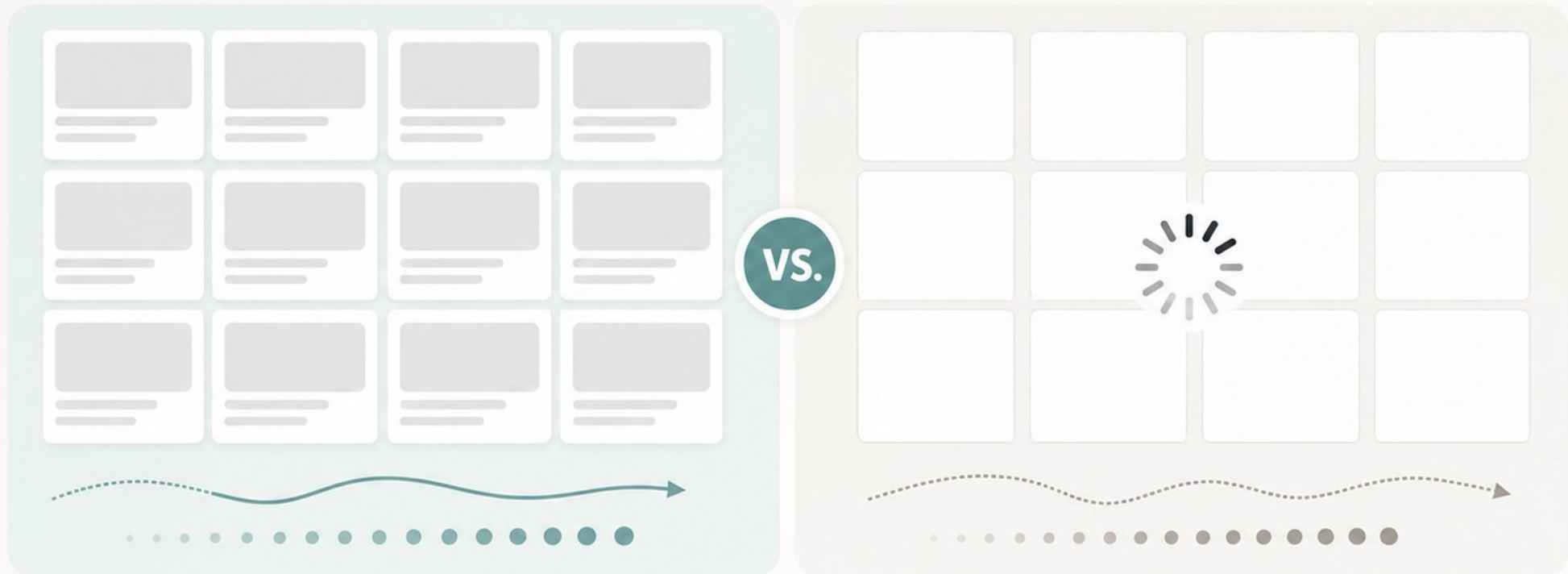
- Avoid nested interactive elements; document whole-card vs button-only actions



- Integrate filtering with visible active filters, result counts, and clear-all



Performance, Perceived Speed, and Loading States



- “Skeleton screens reduce perceived wait by 20–30% when layout is predictable and load time is 400ms–3s”



- “Skeletons must match real content dimensions precisely to prevent Cumulative Layout Shift (CLS)”



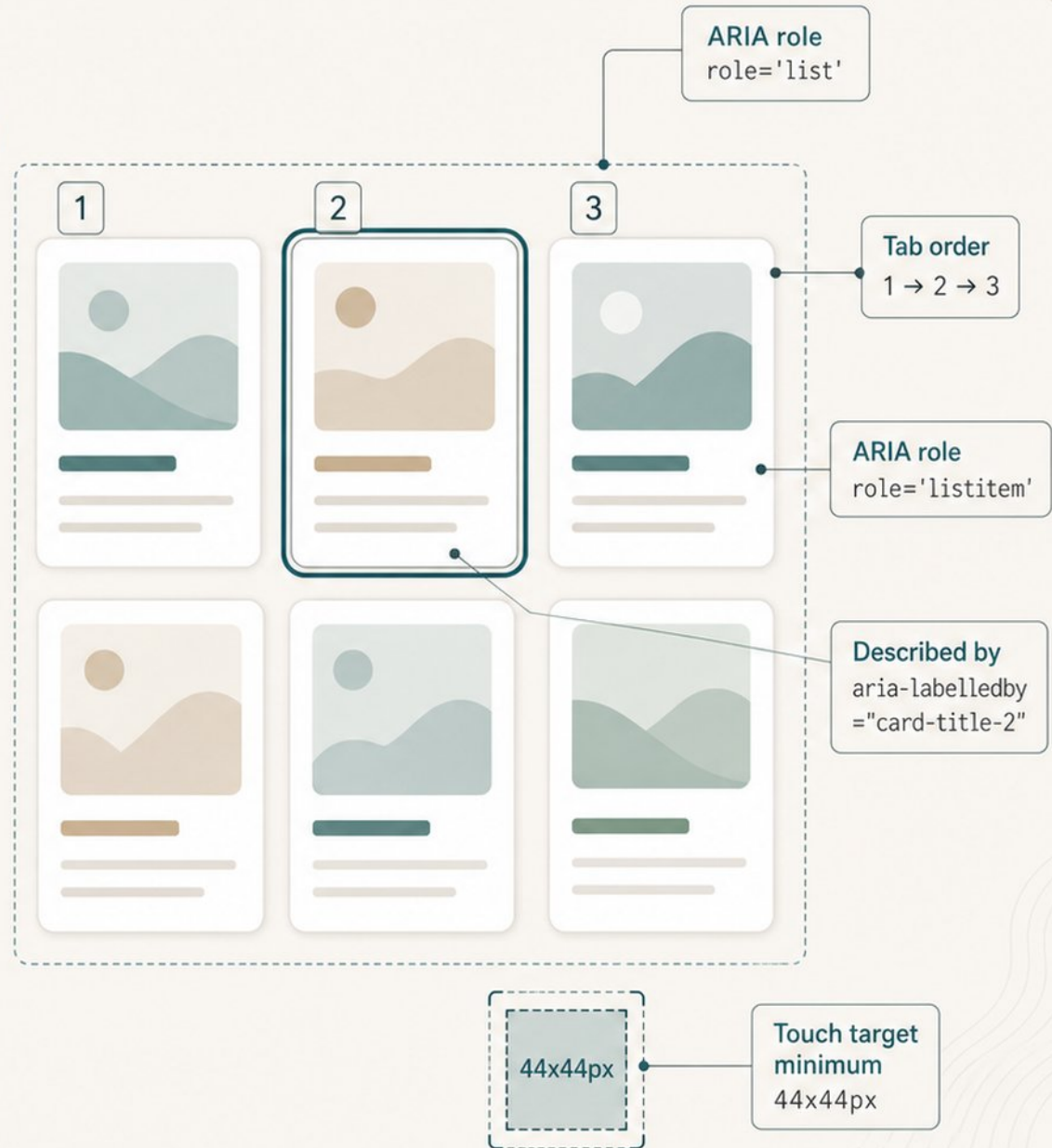
- “Render items as they arrive in a staggered reveal to keep the grid stable and interactive”



- “Disable shimmer animations for users who prefer reduced motion (prefers-reduced-motion)”

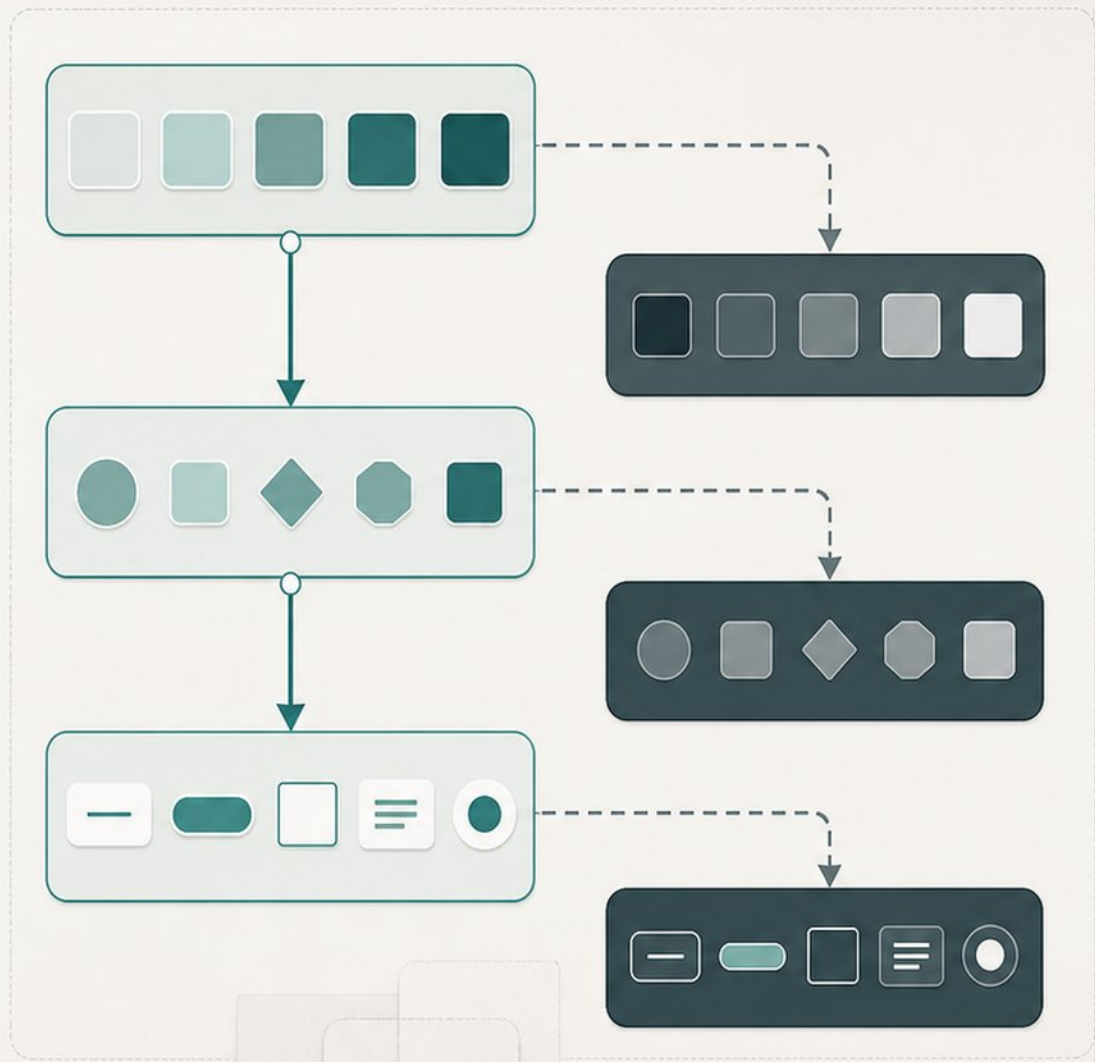
Accessibility in Card Collections

- Use semantic HTML: headings, lists, and landmarks; avoid div-only markup
- Every interactive element must be keyboard-reachable with visible focus rings
- Apply ARIA roles (`role='list'`, `role='listitem'`), `aria-labelledby`, and `aria-busy` for loading
- Touch targets: minimum 44×44px; color contrast: 4:5:1 for text, 3:1 for UI components; never use color alone to convey meaning

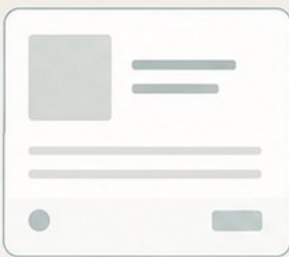


Design System Governance and Token Architecture

- Design tokens are organized in 3 tiers: primitive (raw values), semantic (meaning), and component (scoped properties).
- Components must reference only component tokens, which reference only semantic tokens, which reference only primitive tokens.
- Governance model needs dedicated ownership, clear contribution guidelines, semantic versioning, adoption metrics, and documentation standards.
- Common anti-patterns: hardcoded values, duplicate tokens, component tokens used only once, and missing dark mode variants.

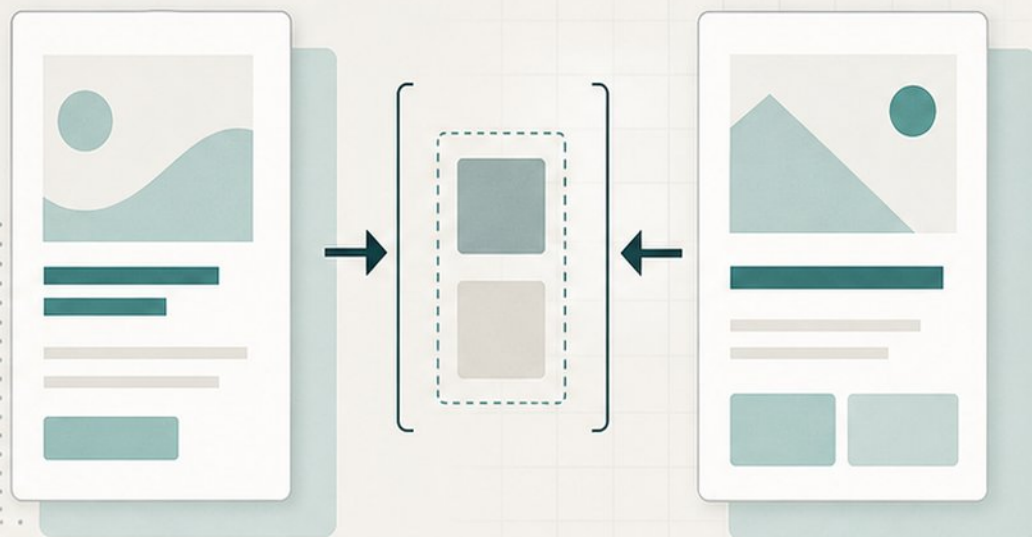


Scalability: Card Variants, Density, and Cross-Platform Consistency

	elevated	outlined	flat
comfortable			
default			
compact			

- Define card variants (elevated, outlined, flat) and density (comfortable, default, compact) as component properties, not separate components.
- Use boolean properties (hasMedia, hasFooter, hasBadge) to extend a single base card component.
- Dark mode: bind surface, border, and shadow tokens to modes; test both themes for contrast compliance.
- Responsive strategy: mobile-first single column, tablet 2 columns, desktop 3–4 columns; use container queries for adaptive card internals.

Testing and Evaluating Card Collections



Use moderated, unmoderated, A/B, first-click, and preference testing for comparison tasks



Track task completion, time-to-complete, error rate, and direct path rate



Validate IA structure with tree testing and card sorting before visual design



Test accessibility: keyboard-only, screen reader, 200% zoom, reduced motion, and color blindness simulation

Card Design Checklist and Common Mistakes



- ✓ One concept per card, summary under 100 characters, one primary action
- ✓ Cramming too much content and inconsistent card structure break scannability
- ✓ Fix jagged heights: truncate text, enforce aspect ratios, or use masonry
- ✓ Avoid nested interactive elements: one link per card, or link in action area

COMMON CARD DESIGN MISTAKES

CRAMMED CONTENT



FIX: Simplify.
Keep it scannable.

JAGGED HEIGHTS



FIX: Align heights.
Truncate or use
masonry.

MISSING FOCUS



FIX: Establish
hierarchy.
Lead with one idea.

NESTED LINKS



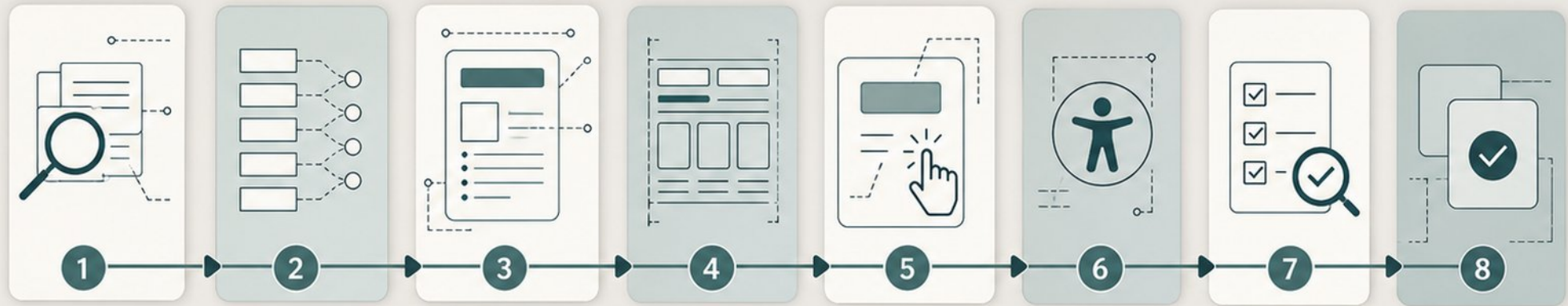
FIX: One link
per card, or link
in action area.

HEAVY SHADOWS



FIX: Use subtle
elevation for a
clean, calm feel.

Putting It All Together: A Design Workflow



Start with user tasks: define questions users must answer before choosing layout, density, and attributes



Build one card template, generate a small batch, print and test, fix the template, regenerate all



Use design system starter kits, Figma templates, token architecture frameworks, and usability testing platforms

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/f6e779ae/public