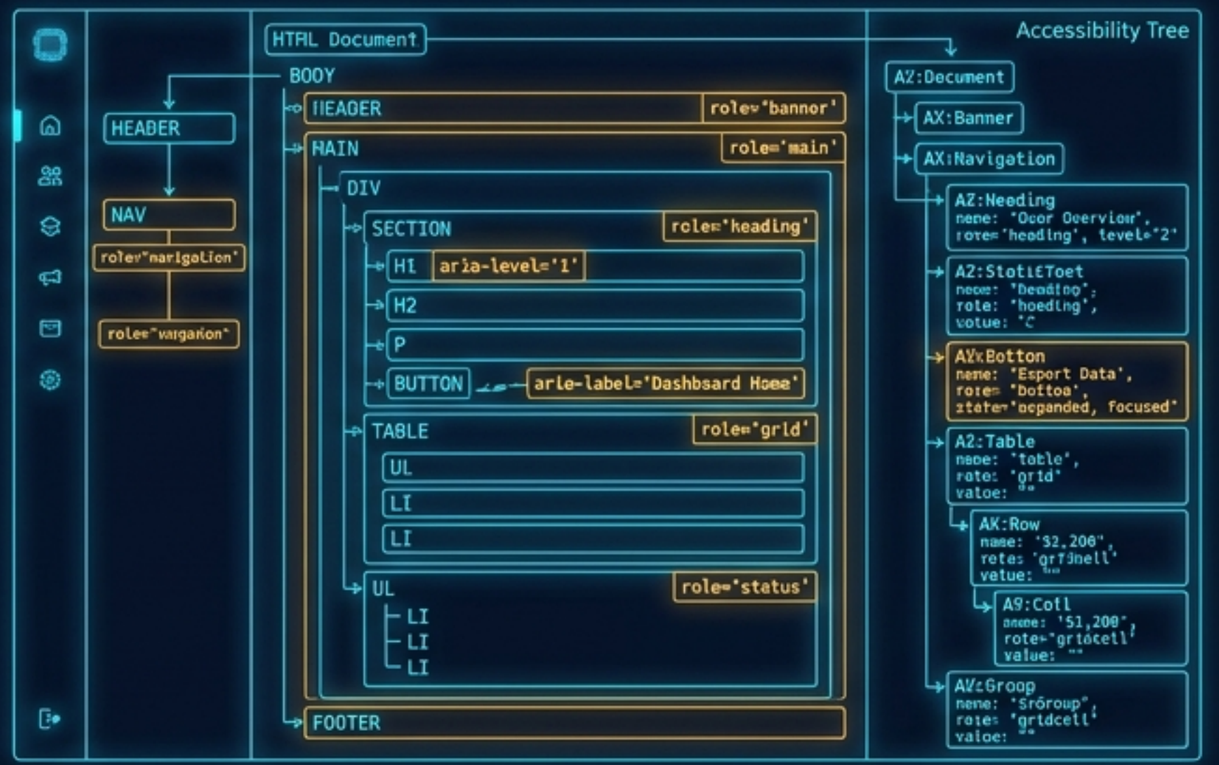
DIGITAL ACCESSIBILITY: S10E01

People, Barriers, and Assistive Technology

Moving from our 5-modality smoke audit in S10E00 into how assistive technologies and browser APIs actually work under the hood.





Deep landmark/heading/accname-1.2 cascades are saved for S10E02. The standalone diagrams of the Accessibility Tree pipeline, Browse vs. Focus Mode, and WCAG 2.2 legal mappings in this deck are designed to double as crisp, printable reference c

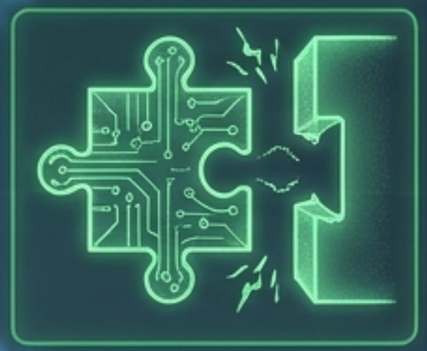
Gemini Notebook

Ch. 1 — Disability Models & Spectrum of Barriers



The Medical Model

- Frames the person as broken or impaired.
- Focuses heavily on fixing the user or building separate, specialized, and often unequal accommodations.



The Social / Interaction Model

- Frames rigid design as the barrier. Disability occurs exclusively at the mismatched interaction between a person and their environment.
- Focuses on flexible, inclusive design that adapts to the user's needs natively.

Gemini Notebook

The Persona Spectrum Matrix

Modality	Permanent	Temporary	Situational
Visual	Blindness / Low Vision	Cataracts / Eye Patch	Severe Sun Glare
Auditory	Deafness	Ear Infection	Bartender in a loud club
Motor / Dexterity	One Arm / Paralysis	Broken Wrist	Holding a baby
Speech	Non-verbal	Laryngitis	Heavy Accent / Noisy Room
Cognitive / Neuro	Down Syndrome	Concussion	Extreme Fatigue / Panic

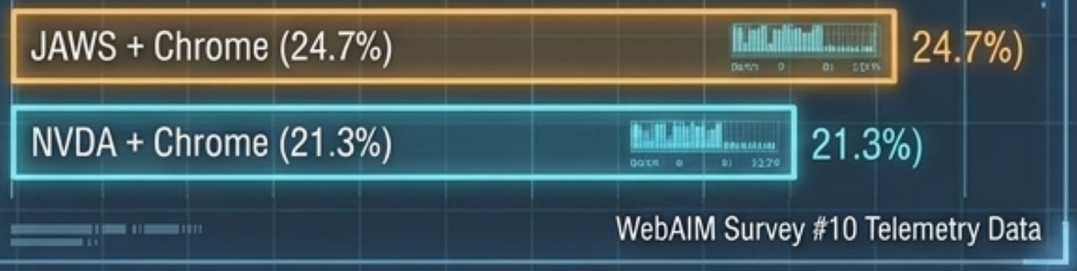
The Universal Benefit: Solving for permanent disabilities inherently provides the technical infrastructure to solve for temporary and situational constraints.

Gemini Notebook

Ch. 2 — Assistive Technologies in Real Daily Use

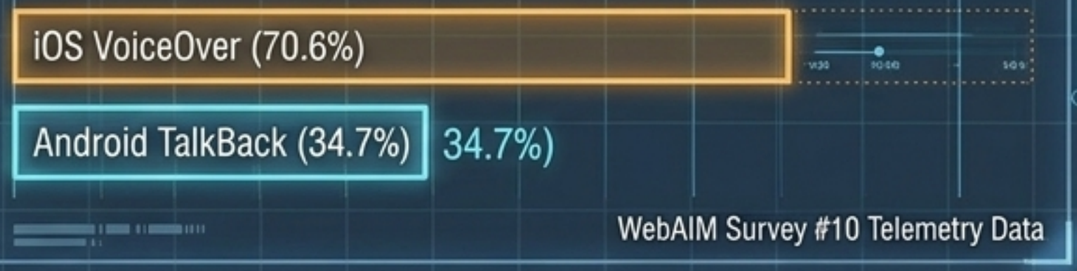
The Screen Reader Landscape

Desktop Dominance



WebAIM Survey #10 Telemetry Data

Mobile Dominance



WebAIM Survey #10 Telemetry Data

Desktop: Browse vs. Focus Mode

Browse Mode: Virtual cursor.
Intercepts keystrokes for structural navigation (e.g., pressing ‘H’ jumps to the next heading).

Focus Mode: Passthrough.
Passes keystrokes directly to the browser for native form interaction.

Mobile: Interaction Gestures



iOS VoiceOver: The Rotor
(Two-finger dial to switch navigation context).



Android TalkBack:
Multi-directional swipes
(e.g., Swipe down then right for global menus).

Gemini Notebook

Visual Adaptation: Magnifiers & Braille

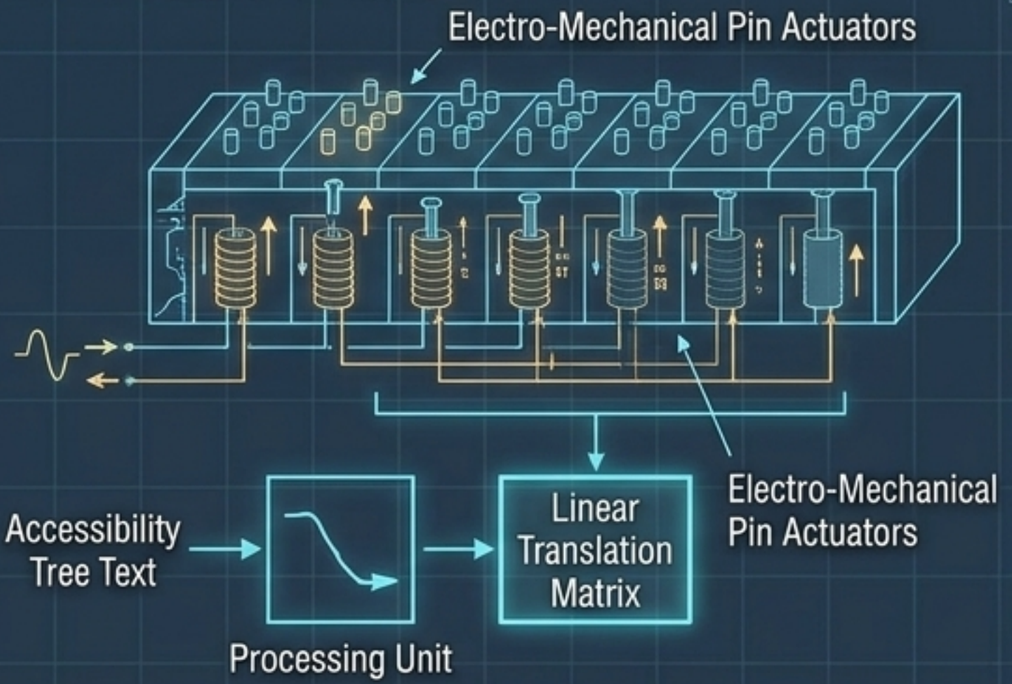
Screen Magnifiers: Diagnostic

Screen Magnifiers (ZoomText / macOS Zoom)



- The Barrier: Loss of peripheral context. Distant UI elements (like top-right notification toasts) disappear completely from the user's field of view.
- The 400% Reflow Solution: Browser zoom triggering a single-column mobile layout (320 CSS px) on desktop screens, eliminating horizontal scrolling.

Refreshable Braille

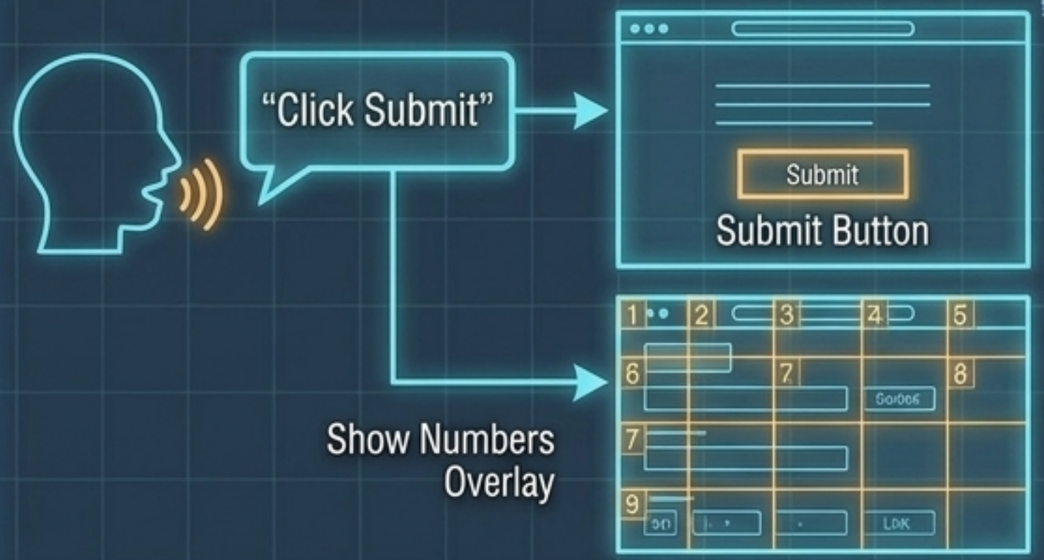


- The Mechanic: Renders the programmatic output of a screen reader linearly, translating the Accessibility Tree text into tactile pins. Requires exact programmatic names and states to function.

Gemini Notebook

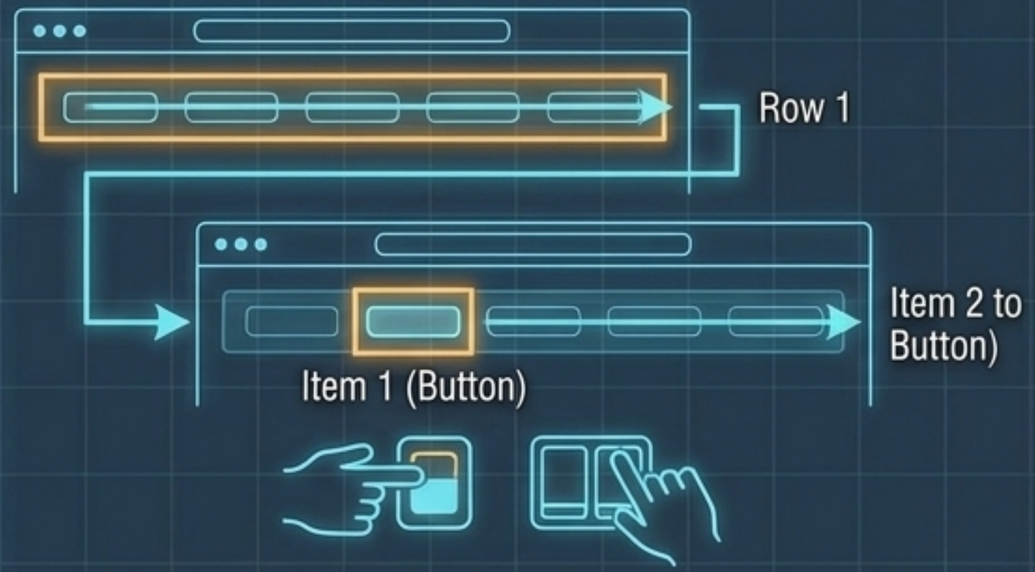
Motor & Speech Navigation: Voice Control & Switches

Speech Recognition (Voice Control / Dragon)



- Mechanic: Maps the spoken word to the visible text label on a control.
- Fallback: If programmatic labels fail, users trigger a Show Numbers overlay grid to select an interactive element.
- Crucial Rule: Voice users rely entirely on visual labels. Hidden or mismatched ARIA labels break activation.

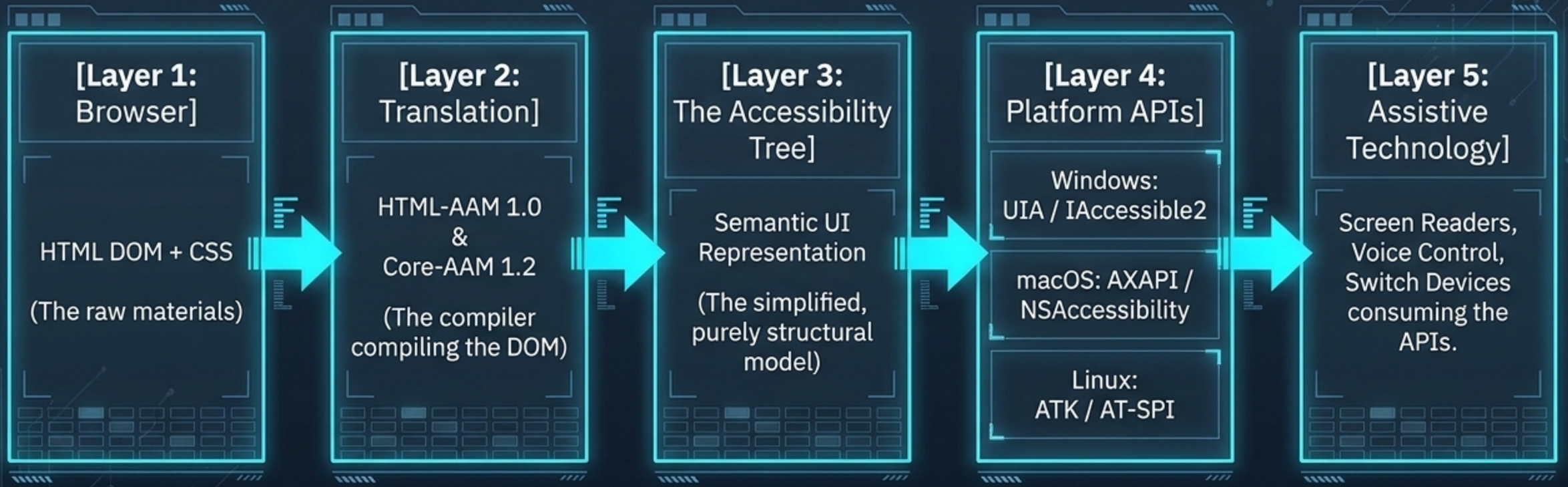
Switch Scanning (Single / Dual Switch)



- Mechanic: Highlights interfaces systematically—first by row, then by individual item.
- The Barrier: Exhaustion. Single/Dual switch users require ruthless focus order and efficiency. Every extraneous or hidden tab stop represents unnecessary physical labor.

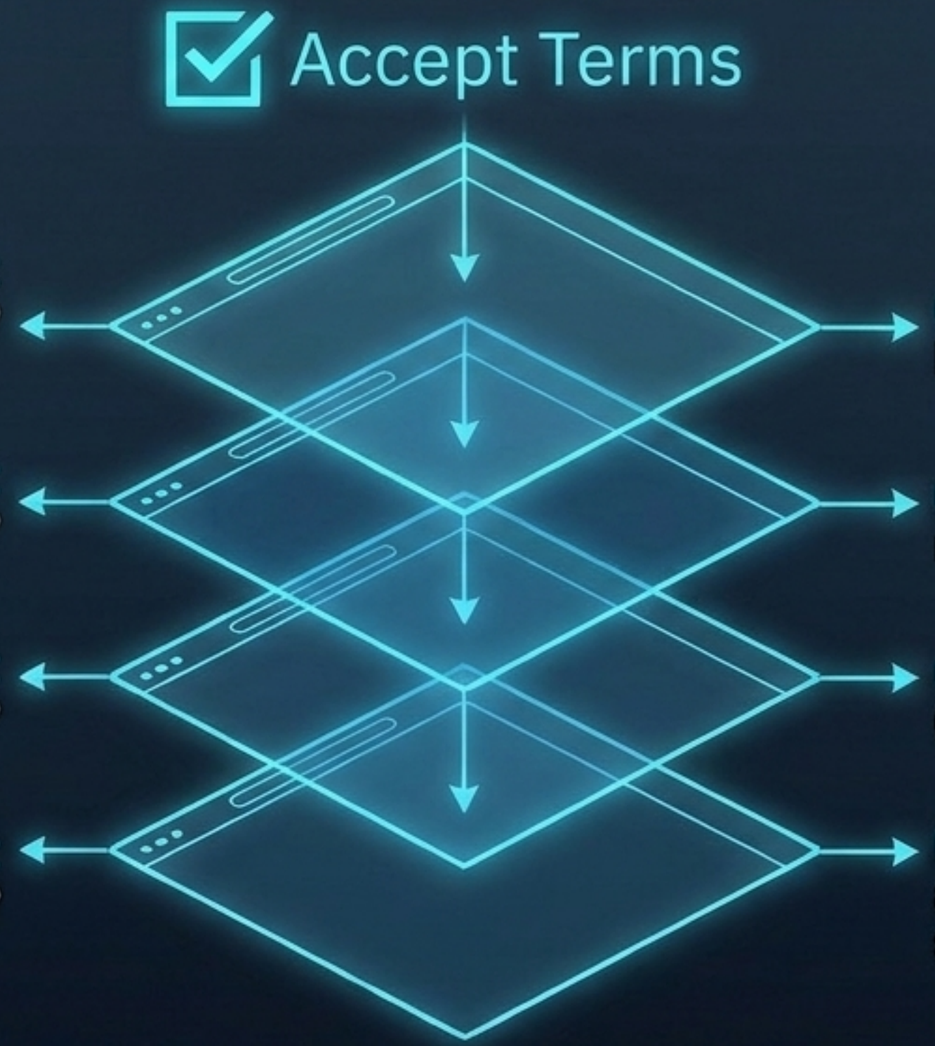
Gemini Notebook

Ch. 3 — Under the Hood: The Accessibility Pipeline



Gemini Notebook

The Anatomy of an Accessibility Node



Layer 1:
ROLE: What is it?

Computed: checkbox (Tells the AT how the user can interact with it).

Layer 2:
NAME: How do we refer to it?

Computed: Accept Terms (The label read aloud or targeted by voice).

Layer 3:
STATE: What condition is it in?

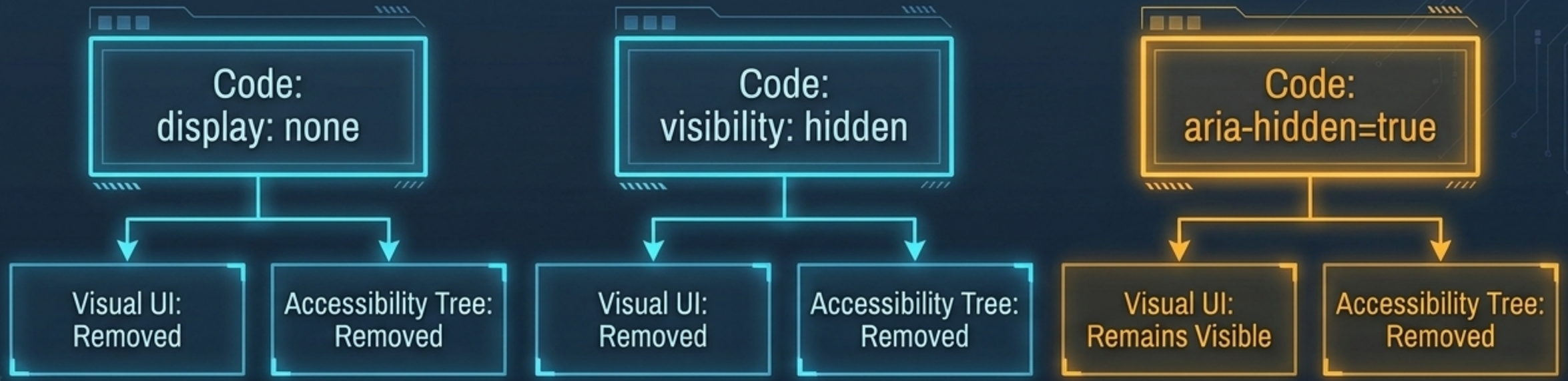
Computed: checked: true (Dynamic boolean conditions).

Layer 4:
VALUE: What is its current data?

Computed: N/A (Not applicable for a checkbox, but would be 50 for a slider or volume for a meter).

Gemini Notebook

Pruning the Tree: Hiding Content



Danger Callout: The Phantom Ghost Stop

Applying `aria-hidden=true` to a natively focusable control (like a `<button>`) creates a critical failure. Keyboard users will Tab to the element (seeing the focus ring), but screen readers will hear hear absolute silence because the node was pruned from the Accessibility Tree.

Gemini Notebook

Ch. 4 — Standards & Legal Conformance Landscape

The Rulebooks: WCAG 2.2 & WAI-ARIA 1.2

WCAG 2.2

- **Framework:** POUR (Perceivable, Operable, Understandable, Robust).
- **Versions:** A backward-compatible cascade (2.0 -> 2.1 -> 2.2).
- **Levels:** A, AA (Standard), AAA (Strict).

WAI-ARIA 1.2 (The Polyfill)

Crucial Rule: ARIA modifies Accessibility Tree semantics ONLY.

⚠ **The Warning:** Adding **role=button** to a ** ** adds ZERO native browser keyboard support (Tab, Enter, Space) and ZERO visual behavior. It is a promise to the screen reader that you have written custom JavaScript to handle the rest.

Gemini Notebook

The Global Legal Mapping Matrix

	Legal Mandate	Jurisdiction	Technical Standard Target
1	US DOJ ADA Title II	United States (State/Local)	WCAG 2.1 / 2.2 AA
2	Revised Section 508	United States (Federal)	WCAG 2.0 AA
3	European Accessibility Act (EAA) / EN 301 549	European Union	WCAG 2.1 AA (Web & Software) + WCAG2ICT
4	AODA	Ontario, Canada	WCAG 2.0 AA

The Unifying Takeaway: Targeting WCAG 2.2 AA serves as the definitive, future-proof technical baseline for global legal compliance.

Gemini Notebook

Ch. 5 — 5-Modality Broken -> Fixed Walkthrough

Modality 1 & 2: Visual & Keyboard (Pathway Desk)

[–] Broken: The Custom

```
<span onclick=filter() class=chip>
```

- Visual Bug: CSS hiding relies on opacity tricks, which fail entirely in Windows High Contrast Mode.
- Keyboard Bug: The **** skips the Tab sequence. Switch users cannot reach it at all. Keyboard users cannot activate it with Enter or Space without extra custom JavaScript.

[+] Fixed: The Native <button>

```
<button aria-pressed=true
class=chip>
```

- Result: Inherits native focus ring, native Enter/Space activation, and native High Contrast support directly from the browser.

Gemini Notebook

Modality 3: Screen Reader (Operations Desk)

[–] Broken Output (Raw DOM):

```
Pathway Lesson 1, group.
Toggle captions, text.
Complete exercise, generic.
```

The Flaw: Missing roles and states. The user does not know 'Toggle captions' is interactable, or whether the exercise is currently expanded.

[+] Fixed Output (Accessibility Tree):

```
Toggle captions, toggle button,
not pressed.
Complete exercise: Accessibility
Tree Quiz, button, expanded.
```

The Fix: Assigning accurate **Roles** (**button**), **States** (**aria-pressed**, **aria-expanded**), and **Names** (**Accessibility Tree Quiz**) turns raw text into an actionable interface.

Gemini Notebook

Modality 4: 400% Zoom (The Distant Toast)

[-]

Broken: Absolute Positioning

A success toast message pops up in the distant top-right corner of the absolute browser window.

Because the user is magnified at 400%, their viewport is focused on the center.
The toast appears entirely outside their field of view.



[+]

Fixed: Inline Context

The status message is rendered inline, immediately adjacent to the trigger button.

Designing for 320px single-column reflow ensures dynamic content updates occur within the immediate visual context of the user's interaction.

Gemini Notebook

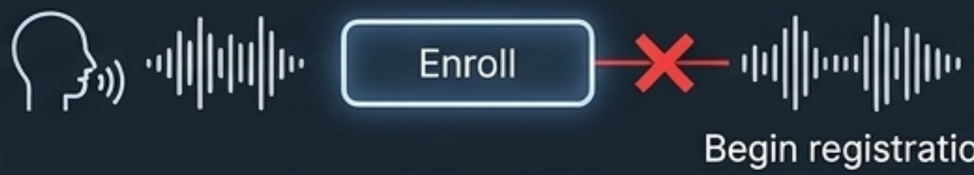
Modality 5: Mobile, Voice & Switch (The Label Mismatch)

Enroll

[-] Broken: The `aria-label` Override

```
Code: <div aria-label=Begin registration>Enroll</div>
```

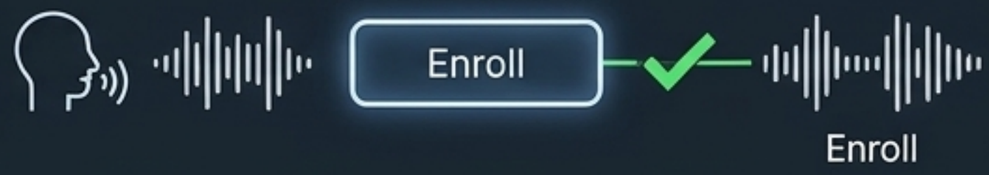
The Flaw: A Voice Control user looks at the screen and says "Click Enroll'. Nothing happens. The Accessibility API name has been completely overwritten by the aria-label to "Begin registration'. The voice engine cannot find "Enroll'.



[+] Fixed: Matching Labels

```
Code: <button>Enroll</button>
```

The Fix: Use a native control where the programmatic Accessible Name matches the visible label exactly, allowing speech recognition algorithms to map spoken commands to interface targets.



Gemini Notebook

Ch. 6 — Annotated Code Diff & DevTools Exercise

The Anatomy of a Fix

Role: Generic

Name: CC

State: None (CSS .active is invisible to AT)

Keyboard: Unreachable

```
[-] <div class=cc-toggle active
      onclick=toggleCC()>CC
    </div>

[+] <button type=button class=cc-toggle
      aria-pressed=true>Captions (CC)</button>
    </button>
```

Role: Button (Native)

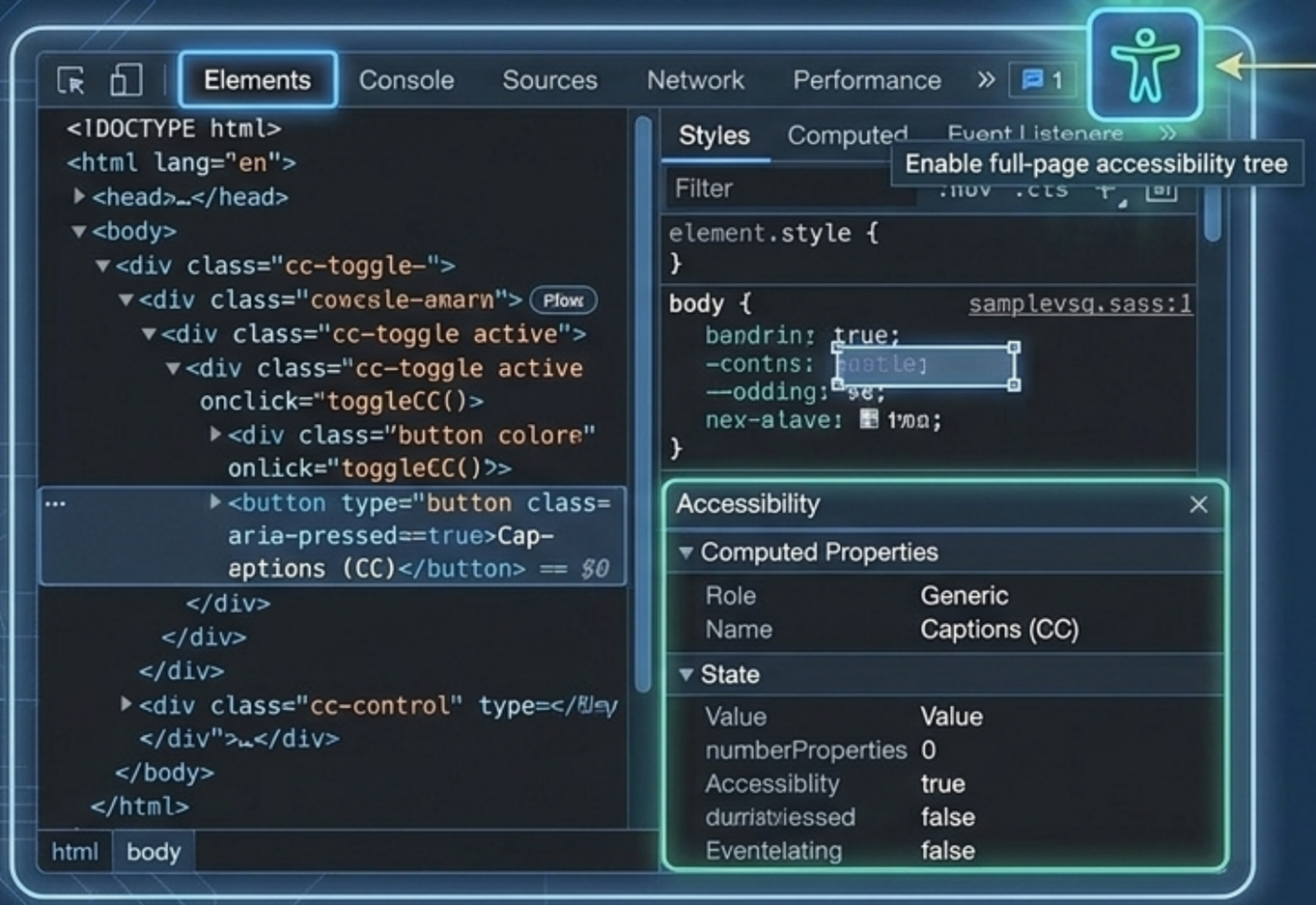
Name: Captions (CC)

State: pressed = true (via ARIA)

Keyboard: Native Tab/Enter/Space

Gemini Notebook

Inspecting the Full Accessibility Tree (FAT) in Chrome



Step-by-Step Exercise:

1. **Enable the Tree**: Toggle on the “Enable full-page accessibility tree” icon in the top right of the Elements panel to switch from DOM view to FAT view.

2. **Inspect the Control**: Target a Pathway or Operations Desk control.

3. **Verify the Blueprint**: Look at the Computed Properties pane. Ensure the browser's computed Role, Name, State, and Value match the intended design blueprint perfectly.

Gemini Notebook

Ch. 7 — Verification & Next Up

The 4-Point Smoke Pass

Verification Checklist (For QA & Dev):

- [x] **Computed Verification:** Accurate Role, Name, State, and Value verified in DevTools.
- [x] **Ghost Stop Check:** Zero instances of aria-hidden=true applied to focusable nodes.
- [x] **Experiential Pass:** NVDA/VoiceOver test + 400% zoom layout test + Voice Control ‘Click <label>’ match test.
- [x] **Legal Mapping:** Explicitly maps to WCAG 2.2 AA (satisfying EAA, ADA Title II, and Sec 508).

Tee-Up: S10E02

Next up: Structure, Semantics and Navigation.

We will dive deep into Landmark hierarchies, Heading cascades, and AccName-1.2 computation rules.

Gemini Notebook