

Silicon UI: Project Handbook

Oct 10, 2026 · @Anil

Summary. Silicon UI is a React 19 and TypeScript component library of about 106 components, built by porting ArcUI (github.com/kuratlielia/arc-library, MIT licence, docs at uiarc.dev) so that every component matches Arc exactly in motion, typography (size, weight, colour) and control heights, while spacing and corner radii come from Silicon's own tokens. A script-driven pipeline captures Arc from its live docs site into JSON specs, fresh AI agents then port one component each, and five automated checks decide pass or fail, never the AI's own judgment. This handbook gives the terms, the pipeline, exact commands, the decisions made and why, known problems and open work. Read sections 1 to 4 for context, and the rest if you will run or extend the pipeline. Status date: 2026-10-10.

1. Overview

Silicon UI is the component library of Team of Silicons. It is a port of ArcUI, an open-source React component library, into the team's own design system. The goal is that a screen built with Silicon UI moves, reads and feels like Arc, but sits on Silicon's spacing and radius scales.

The problem. Porting about 106 components by hand is slow and drifts from Arc over time. Letting AI agents port them unchecked is worse, because an agent will say a component matches when it does not. The pipeline replaces judgment with measurement: Arc is measured once, and every port is measured against that record.

Who it is for.

- **Designers and engineers at Team of Silicons** who build screens from the library. The project owner is Anil Choudhary, Head of Design.
- **AI agents** that port components and need the rules, files and commands in one place.
- **Screen-design agents** (future) that will need a per-component descriptor file. See section 11.

Status on 2026-10-10. All 106 Arc specs are captured and the spec audit passes (clean 106, mismatched 0). Ported and passing the checks: search-field and breadcrumb. Existing before the pipeline: input and badge. All other components are in the state

`preflighted`, which means the spec is ready and the port has not run yet. `text-reveal` was recaptured and a re-port was started; its result is [TO CONFIRM]. Nothing is merged into the main migration branch yet.

2. Scope

What this covers.

- Porting about 106 Arc components into Silicon UI (React 19, TypeScript strict, CSS modules, Radix, Storybook, Next.js consumer in `apps/web`).
- The capture pipeline that records Arc's real rendered styles and motion from `uiarc.dev` into `specs/<name>.json` .
- The token rules that map Arc's spacing and radii onto Silicon's scales.
- The five automated gates that accept or reject a port, and the driver script that runs everything.
- Operating the pipeline: commands, resuming after a stop, reading the report, handing work to a new session.

What it deliberately does not cover.

- **Font family.** Only font size, weight and colour must match Arc. Silicon keeps its own font family.
- **Spacing and radii copying.** Arc's values are snapped to Silicon tokens, not reproduced. A port that is 2px wider than Arc because of a token is correct.
- **JavaScript-driven motion.** Scripts cannot verify it (for example the stagger in `text-reveal` or the `clear-button` motion in `search-field`). A person reviews these by eye in Storybook.
- **Linux or cloud rendering parity.** Fonts differ on Linux and this is untested.
- **Distribution.** Whether the library ships as an npm package or as copy-in files is undecided. See section 11.
- **Merging.** The trial branches are not merged into `react-migration` yet.

3. Key concepts and terms

Term	Meaning
ArcUI / Arc	The open-source React component library being ported (github.com/kuratlielia/arc-library , MIT, docs at uiarc.dev). Its source is also kept locally under <code>reference/arc-library</code> .
Silicon UI	This project. Arc's components rebuilt on Silicon tokens.
Silicon tokens	Named spacing and radius values that replace Arc's raw pixels (scales in section 10).
Spec	A JSON file <code>specs/<name>.json</code> holding Arc's measured styles for one component, in light and dark themes. It is the single source of truth a port is checked against.
Variant	One configuration of a component inside a spec, keyed under <code>themes.light</code> or <code>themes.dark</code> , for example <code>reveal</code> .
Part	A styled element inside a variant (for example <code>label</code> , <code>item</code>). Siblings of the same kind are <code><part>@<n></code> , such as <code>item@1</code> .
State	An interaction state of a part (hover, focus, active). A target that is collapsed is recorded as collapsed.
Root	The top element of a variant. It must be a class that exists in Arc's own CSS module.
Dotted root	When one variant has two genuinely different instances, the second is keyed <code>root.1</code> , then <code>root.2</code> , in DOM order. Identical repeats collapse.
Chrome (demo chrome)	Wrapper elements that only exist on the uiarc.dev docs page (classes such as <code>demo</code> , <code>card</code> , <code>stage</code>). They are not part of Arc's component and are filtered out.

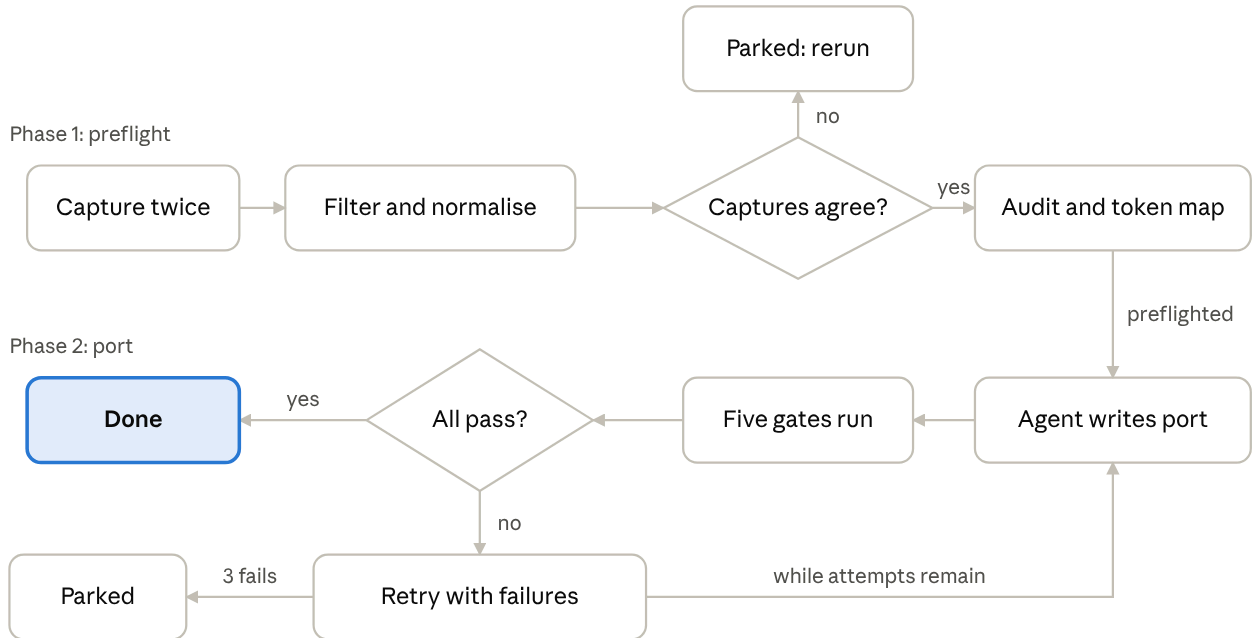
Term	Meaning
chromeHints	A per-variant note such as <code>reveal -> title</code> giving the docs page's own name for an instance. Naming help only. No check reads it.
Isolation	During capture, demo classes are removed from every captured element at once, measured, then restored, so the spec records only Arc's own CSS.
Preflight	Phase 1 of the driver. Captures Arc twice, requires the two captures to agree, builds the token map and writes the spec. Makes no AI calls.
Port	Phase 2. A fresh AI agent writes the component, then the five gates run.
Gate / check	An automated pass or fail script: <code>check-tokens</code> , <code>check-motion</code> , <code>typecheck</code> , and <code>check-component</code> in light and dark.
Driver	<code>scripts/driver.mjs</code> , which runs preflight and port per component and writes <code>REPORT.md</code> and <code>driver-state.json</code> .
Statuses	<code>preflighted</code> (spec ready), <code>done</code> (port passed), <code>existing</code> (built before the pipeline), <code>parked</code> (stopped, needs a person), <code>needs-recapture</code> (spec must be captured again), <code>stopped</code> .
Parked	The driver gave up on a component and listed the reason in <code>REPORT.md</code> .
NEEDS-DECISION	A port agent's report that the spec asks for something Arc's own CSS does not do. The driver parks immediately when the same checks fail twice with this report.
Computed override	An entry in <code>tokens-overrides.json</code> that pins a CSS expression to specific properties instead of snapping to a token (for example text-reveal's <code>.clip</code> em values).

Term	Meaning
Deviation	An approved, recorded difference between Arc and Silicon in <code>deviations.json</code> (for example the focus ring on <code>focus</code> and <code>active</code>).
Respec / accept-respec	Publishing a recapture of an already-ported component, allowed only when two named content hashes match.
Canonicalization	Colours from both sides are converted to OKLab (using <code>culori</code>) before comparing, so equal colours written differently compare equal.
parentPart	A field on each part in newly captured specs naming the part it inherits from. Used by the inheritance check.
Review by eye	Items scripts cannot verify, listed in <code>REPORT.md</code> for a person to check in Storybook.

4. How it works

The driver runs two phases per component. Preflight turns Arc into a trusted spec. Port turns the spec into a component and proves it matches. A person only gets involved when something is parked.

Preflight turns Arc into a spec, then five gates decide every port



Parked components wait for a person. The reason is written to REPORT.md.

driver pipeline · 2 phases, 2 decisions

Preflight uses no AI. A component that fails its gates is retried, and is parked once its three attempts are used.

Phase 1: preflight (no AI, costs time only)

1. **Capture twice.** A Playwright browser opens the component's page on uiarc.dev and records computed styles for every part and state, in light and dark. `scripts/capture-config.json` says how to open overlays and which element is the root.
2. **Normalise.** `scripts/normalize-spec.mjs` drops docs-page chrome (any element whose classes Arc's own `<name>.module.css` does not define), keeps real instances under dotted roots, canonicalises colours and writes the spec.
3. **Require agreement.** The two captures must produce the same themes. If they differ the component is parked.
4. **Spec-root audit.** `scripts/audit-spec-roots.mjs` checks that every part and root class in the spec exists in Arc's CSS module. A spec that fails cannot be marked verified.
5. **Token map.** Each raw value is mapped to a Silicon token, or flagged for a decision (section 10).

Result: status `preflighted` .

Phase 2: port (AI agent, then gates)

1. The driver starts a **fresh agent** with the spec, `AGENTS.md` , the token map and Arc's source for reference. The agent may only change files in `src/components/<name>/` .
2. The five gates run (table below).
3. If a gate fails the agent gets the failures and tries again, up to **3 attempts**.
4. Pass gives `done` . Failing every attempt, or reporting NEEDS-DECISION twice on the same failing checks, gives `parked` .

The five gates

Gate	What it checks	Notable rules
<code>check-tokens</code>	Spacing and radius values in the port use Silicon tokens.	Exact match, else nearest token, tie rounds up. More than 2px from any token needs an entry in <code>tokens-overrides.json</code> . Multi-value shorthands (gap, padding, margin, border-radius, inset) are mapped per axis. The visually hidden block is exempt verbatim (<code>-1px</code> stays <code>-1px</code>).
<code>check-motion</code>	Durations, easing and animation properties match the spec.	JS-driven motion is outside its reach.
<code>typecheck</code>	<code>tsc --noEmit</code> passes.	Strict TypeScript.
<code>check-component (light)</code>	The rendered component's computed styles match the spec in the light theme.	Finite animations and transitions are drained (cap 5s) before measuring. Auto-sized containers skip height and min-height when Arc's CSS gives that class none. The focus ring deviation applies to <code>active</code> and to any state where <code>:focus-visible</code> matches.
<code>check-component (dark)</code>	Same, in the dark theme.	Same rules.

What the driver adds around this

- Resumable: `--resume` skips components already `preflighted` or `stopped`, so a stopped run continues where it left off.
- Reports to `REPORT.md` (with `## DECISIONS NEEDED` and, after a usage-limit stop, `## RESUME`) and state to `driver-state.json`.
- Limits: 20 minutes per capture by default (a `timeoutMs` of 2400000, 40 minutes, is set for usage-meter, time-picker and swipe-actions), a port breaker at 4 failures, and `--max-hours` for the whole run.

- **Network resilience:** navigation retries HTTP 5xx and 429, and known transient `net::ERR_*` errors, on a 5s, 15s, 45s ladder. A health probe of `https://uiarc.dev` runs once before a preflight phase. If the site is down the run stops without spending any component's attempts.

5. Setup and usage

All commands run in Terminal (Warp is used on the project machine) from the repo folder. Installing dependencies is not recorded in the project history: [TO CONFIRM] whether `npm install` is all that is needed.

1. Go to the repo and check the branch and tree.

```
cd /Users/anil/Documents/silicon-ui-react && git branch --show-current &&
git status --short | head
```

Expected: `port-trial-2` and no changed files. If the tree is dirty, stop and find out why before running anything.

2. Check whether the driver is already running.

```
ps aux | grep -c "[d]river.mjs"
```

`1` means running. `0` means not running (it does not mean stuck).

3. Check the site is up before a preflight.

The driver now probes `uiarc.dev` itself, but a manual check is cheap. This command is an example, not a project script:

```
curl -s -o /dev/null -w "%{http_code}\n" https://uiarc.dev
```

Expected: `200`.

4. Run preflight for chosen components.

Keep the laptop plugged in and on one Wi-Fi network. `caffeinate` stops it sleeping.

```
cd /Users/anil/Documents/silicon-ui-react && caffeinate -dimsu node
scripts/driver.mjs \
  --phase preflight --branch port-trial-2 --max-hours 2 --resume \
  --only in-view-title,resizable-panels,scroll-area,slot-text,text-
reveal,text-shimmer \
  2>&1 | tee -a /tmp/recapture-6b.log
```

`--resume` skips anything already `preflighted`. To recapture a component that is already `preflighted`, its status must first be set to `needs-recapture` in `driver-state.json`. How this was done is not recorded: [TO CONFIRM].

5. Read the result. Each component ends with one line:

```
grep -E "^(in-view-title|text-reveal): " /tmp/recapture-6b.log | tail -12
```

Expected line: `text-shimmer: preflighted - Two captures agree; token map complete, then STOP: plan complete`. Details are in `REPORT.md` and under `runs/<run-id>/`.

6. Run the port phase.

```
node scripts/driver.mjs --phase port --branch port-trial-2 --only text-reveal
```

Overnight batches use `--phase port --branch <branch> --push --resume --limit <n>`. The exact meaning of `--limit` and whether `--push` pushes per component are [TO CONFIRM].

7. Verify the whole repo.

```
node scripts/audit-spec-roots.mjs
npx tsc --noEmit
```

Expected: audit exit code 0 with clean 106, mismatched 0, cannot-check 0; `tsc` prints nothing. The driver selftest and node test commands are [TO CONFIRM] (last known: 39 driver selftests, about 74 node tests, 26 capture tests, all passing).

8. Review by eye. Start Storybook, open each ported component, and check what `REPORT.md` lists under review by eye (JS motion, enter and exit animations, reduced motion).

```
cd /Users/anil/Documents/silicon-ui-react && npm run storybook
```

The Next.js consumer app is served at `localhost:3007` [TO CONFIRM the start command].

9. Accept a respec (only for an already-ported component whose recapture differs). Run `--accept-respec <name>` with the from and to themes hashes written in `runs/<run-id>/<name>/recaptured.json`. It refuses unless both hashes match exactly. The exact hash flag names are [TO CONFIRM]; the record is stored under `respec` in `driver-state.json`.

If a run stops (Ctrl+C, network drop, usage limit, sleep): run the same command again with `--resume`.

6. Examples

Example 1: a clean preflight

Rerunning only text-shimmer after a comparison fix:

```
cd /Users/anil/Documents/silicon-ui-react && caffeinate -dimsu node
scripts/driver.mjs \
  --phase preflight --branch port-trial-2 --only text-shimmer --max-hours 1 -
  -resume \
  2>&1 | tee -a /tmp/recapture-text-shimmer.log
```

Output ended with `ELAPSED text-shimmer: 59.38s`, then `text-shimmer: preflighted` - Two captures agree; token map complete, then `STOP: plan complete`. The WARN lines about drop target ... demo chrome are normal: they list docs-page wrappers the filter removed.

Example 2: edge case, two different instances in one variant

text-reveal's docs page shows two instances of the same Arc class `reveal`: a 52px title and a 16px lead paragraph. Early versions of the filter dropped the second one and silently lost 8 real Arc style sites. The fixed normaliser keeps both:

- `reveal` (hint `title`) and `reveal.1` (hint `lead`) in both themes.
- `reveal.1`'s `clip` records `padding-top: 1.6px`. That value is the evidence for the computed override that pins `.1em` to padding-top only.
- If a third distinct instance existed it would be `reveal.2`. Identical repeats collapse into one. If a collision cannot be resolved, capture throws instead of dropping anything.

Example 3: edge case, captures disagree only on animation timing

text-shimmer parked with "Two captures have different themes". Nine differences were all `background-image` on `shimmer.1`'s `label`. The gradient angle (`90deg`) and the nine colours were identical in both captures. Only the sweep offsets moved, because the shimmer animation was caught at different moments. In one capture the last stop sat at

exactly zero, which Chromium prints as `0%` instead of `calc(0% + 0px)` . The repeatability comparison now masks sweep offsets for this case. A recoloured or re-angled gradient still differs. A rerun then passed.

Example 4: a port that passes, and one that parks

Component	Result	Attempts	Time	Cost (API-equivalent)
breadcrumb	done, all 5 gates	1	83.42s	\$0.5564
search-field	done	1	132.60s	\$0.7373
text-reveal (first attempt at the spec)	parked	2	720.76s	\$1.7517

text-reveal parked because the spec was wrong, not the port: the capture had kept the 52px a demo class added, while Arc’s own CSS declares no font size. The port agent reported NEEDS-DECISION and refused to invent classes to force a pass, which is the correct behaviour. The capture was fixed (isolation) and the component recaptured.

7. Decisions and rationale

Decision	Why	Rejected alternative and why
Scripts decide pass or fail, never the AI.	An agent will claim a match it has not achieved. Measurements cannot be argued with.	AI self-review: unreliable and cannot run unattended overnight.
Capture Arc from the live docs site, twice, and require agreement.	Gives real rendered values and exposes flaky or timing-dependent captures.	A single capture: cannot tell a real value from a mid-animation one.

Decision	Why	Rejected alternative and why
Filter docs-page chrome by checking classes against Arc's own CSS module.	53 of 106 early specs had the docs wrapper (<code>demo</code> , <code>card</code> , <code>stage</code>) as their root.	Migrating those 53 specs in place: impossible, since specs held no ancestry and 50 of 53 composed or computed their class names. They were recaptured instead.
Keep distinct instances under dotted roots (<code>root</code> , <code>root.1</code>). Collapse only identical styles.	The first filter silently lost real Arc sites in 11 components (for example inline-edit lost 20).	Dropping the colliding instance: unsafe coverage loss.
Isolation measures all demo-classed elements together, then restores them.	Measuring one at a time produced a 16px parent with 52px children, which no port could satisfy.	One-element-at-a-time isolation; or marking those properties "unverified", which leaves a hole in the checks.
Spacing and radii snap to Silicon tokens (exact, else nearest, tie rounds up). Values over 2px away need a recorded decision.	Keeps the library on one scale.	Copying Arc's raw pixels: defeats the purpose of Silicon tokens.
Computed overrides for a few expressions (text-reveal <code>.clip</code> em values, bottom-sheet <code>padding-bottom</code>).	Those values are expressions, not design spacing. The bottom-sheet <code>160px</code> is a drag-overscroll extension.	Snapping them to tokens: would change behaviour.
Compare <code>line-height</code> as a ratio to font size.	Arc's page uses a unitless <code>1.4</code> , which scales per element.	Comparing pixel values: flags correct specs.
Inheritance check is report-only for now.	It is sound only for specs that record <code>parentPart</code> . 100 specs do not have it yet, and in-view-title raised a false alarm (an uncaptured <code>h2</code>).	Wiring it as a gate now: would block good specs. Recapturing all 106 just to populate it: hours on a flaky site.

Decision	Why	Rejected alternative and why
Breadcrumb renders plain <code><a></code> , not <code>next/link</code> .	A shared library should not depend on Next.js.	Importing <code>next/link</code> .
Park immediately on an identical NEEDS-DECISION failure.	A third identical attempt wastes money and cannot succeed.	Always using all 3 attempts.
Preflight circuit breaker off, port breaker at 4.	Preflight makes no AI calls so failures cost only time.	Same breaker for both.

Process choices. A fresh agent per component keeps context small and failures independent. The driver is resumable because overnight runs hit Claude usage limits. Cost figures are API-equivalent, taken from Claude Code transcripts.

8. Known issues and limitations

- **uiarc.dev is flaky.** There have been clusters of HTTP 504s and `ERR_CONNECTION_CLOSED` , and a Wi-Fi change gives `ERR_NETWORK_CHANGED` . Retries and a health probe reduce the damage, but a rerun after a pause is sometimes still needed.
- **JS-driven motion is not verified by scripts.** Search-field's clear-button motion, text-reveal's stagger and breadcrumb's enter and exit animation need a human look in Storybook.
- **Cloud and Linux parity is untested.** Fonts differ on Linux, so a cloud run could produce different measurements. A canary is needed before relying on it.
- **Inheritance check covers only 6 specs.** Only text-reveal, slot-text, text-shimmer, in-view-title, resizable-panels and scroll-area were recaptured with `parentPart` . The other 100 are skipped and counted. Closing the gap needs a `directParent` field and recaptures.
- **Old specs for button and tabs.** Both are recorded `preflighted` rather than `done` . Cosmetic.
- **Slow captures.** slider took 1167s, scroll-area about 7 minutes per capture pair, and usage-meter, time-picker and swipe-actions get a 40 minute limit.
- **Claude usage limits.** A long run will stop on "You've hit your session limit". The driver writes `## RESUME` .

- **Capture quirks that are config, not code.** Overlay selectors must point at content, not Radix scrims (`aria-hidden`). Tooltip needs `dismissOnPress` . Hover-card needs pointer leave and enter. Mention-input needs real `@` keystrokes.
- **Port agent tool limits.** It loses calls on commands prefixed with `cd` or containing shell variable assignments. Its prompt tells it the working directory is already the repo root.
- **Token snapping changes the look slightly.** Breadcrumb's 6px gaps became 8px, and the 7px crumb radius became 8px.
- **Only simple components have been ported so far.** Overlays, charts and motion-heavy components are untested end to end.
- **Snapshot risk.** Numbers in this document were last confirmed on 2026-10-09 and the text-reveal re-port result is unknown.

9. Troubleshooting

Symptom	Cause	Fix
Component parked with "Capture/normalization failed (exit 1)", HTTP 504 or <code>ERR_CONNECTION_CLOSED</code> .	uiarc.dev or the network flaked.	Check the site returns 200, wait a few minutes, rerun the same command with <code>--resume</code> .
<code>--resume</code> preflight selects nothing.	The components are already <code>preflighted</code> , and resume skips them.	Set their status to <code>needs-recapture</code> first [TO CONFIRM how], then rerun.
Parked: "Two captures have different themes".	Animation caught at different moments (sweeps, gradients), or a real difference.	Rerun once. If it repeats, diff the two captures in <code>runs/<run-id>/<name>/</code> . Timing-only differences belong in the repeatability masking, not in the spec.

Symptom	Cause	Fix
Parked: "Recapture differs from committed themes after color canonicalization".	The component already has code in <code>src/components/<name></code> , so a recapture is compared instead of published.	Review the difference. If it is intended, run <code>--accept-respec <name></code> with the two hashes.
Port parked with NEEDS-DECISION on the same checks twice.	The spec asks for something Arc's CSS does not do.	Do not force a pass. Read the agent's SUSPECTED CHECK PROBLEMS, fix the spec or capture, recapture, port again.
<code>check-tokens</code> fails on a value more than 2px from a token.	No token is close enough.	Add a decision to <code>tokens-overrides.json</code> (a computed override if it is an expression).
Overlay or popover never opens during capture.	The overlay selector points at a Radix scrim.	Point <code>overlaySelector</code> in <code>scripts/capture-config.json</code> at the content element. Add <code>dismissOnPress</code> if needed.
Run stopped with "You've hit your session limit".	Claude usage cap.	Wait for the reset and rerun with <code>--resume</code> . See <code>## RESUME</code> in <code>REPORT.md</code> .
<code>ps aux grep -c "[d]river.mjs"</code> prints 0.	The driver is not running.	Not a hang. Check <code>REPORT.md</code> and the end of the log for why it stopped.
Run died when the laptop slept or Wi-Fi changed.	Sleep or network change.	Use <code>caffeinate -dimsu</code> , stay on one network, rerun with <code>--resume</code> .

Symptom	Cause	Fix
Audit reports a spec mismatch.	A spec part or root class is not in Arc's CSS module.	Recapture that component. Do not edit the spec by hand.
A Claude Code agent chat goes quiet.	The agent stalled.	Send <code>continue</code> and tell it to commit work in progress to the branch first.

10. Reference

Locations

Item	Value
Repo folder	<code>/Users/anic/Documents/silicon-ui-react</code>
Remote	<code>teamofsilicons/silicon-ui-library</code> (GitHub)
Upstream library	<code>github.com/kuratlielia/arc-library</code> (MIT), docs at <code>uiarc.dev</code> , local copy in <code>reference/arc-library</code>
Branches	<code>react-migration</code> (main migration branch, untouched by trials), <code>port-trial-1</code> , <code>port-trial-2</code> (current)
Older separate repo	<code>/Users/anic/Documents/silicon-ui</code> , branch <code>overnight/2026-10-07-0121</code>
Latest commit seen	<code>6d174abc6c77d7e49bb9508ff70211d371973fdf</code> on <code>port-trial-2</code> (newer commits may exist) [TO CONFIRM]
Ported components live in	<code>src/components/<name>/</code> (the only place a port agent may write)
Consumer app	<code>apps/web</code> (Next.js), served at <code>localhost:3007</code> ; Storybook via <code>npm run storybook</code>

Files

File	Purpose
<code>specs/<name>.json</code>	Captured Arc spec per component (themes, variants, parts, states). 106 files.
<code>scripts/driver.mjs</code>	The driver: preflight and port phases, <code>--resume</code> , reports, breakers.
<code>scripts/capture-arc.mjs</code> , <code>scripts/capture-site.mjs</code>	Capture engine and navigation with retries (<code>navigateWithSiteRetries</code>).
<code>scripts/normalize-spec.mjs</code>	Chrome filter, dotted roots, <code>chromeHints</code> , <code>parentPart</code> .
<code>scripts/audit-spec-roots.mjs</code>	Checks spec classes exist in Arc's CSS module. Exit 0 means clean.
<code>scripts/check-tokens.mjs</code> , <code>check-motion.mjs</code> , <code>check-component.mjs</code>	Three of the five gates (the other two are <code>tsc --noEmit</code> and the light and dark runs of <code>check-component</code>).
<code>scripts/capture-config.json</code>	Per-component capture settings (see below).
<code>tokens-mapping/tokens-overrides.json</code>	Recorded token decisions and computed overrides.
<code>tokens-map.json</code> , <code>deviations.json</code>	Raw value to token map, and approved deviations (location [TO CONFIRM]).
<code>AGENTS.md</code>	Rules for port agents (computed overrides, dotted-root convention, <code>chromeHints</code> , inherited typography).
<code>CHECKS.md</code>	What each check does, including the inheritance self-consistency section.
<code>CAPTURE-RESULTS.md</code>	Record of capture results.
<code>REPORT.md</code>	Run report: parked reasons, <code>## DECISIONS NEEDED</code> , <code>## RESUME</code> , review by eye.
<code>driver-state.json</code>	State for all 106 components (status, preflight status, hashes, <code>respec</code> records).

File	Purpose
<code>runs/<run-id>/<name>/</code>	Per-run logs and artifacts, including <code>recaptured.json</code> .

Driver parameters

Flag	Meaning
<code>--phase preflight</code> or <code>--phase port</code>	Which phase to run.
<code>--only a,b,c</code>	Restrict to named components.
<code>--resume</code>	Skip components already <code>preflighted</code> or <code>stopped</code> .
<code>--branch <name></code>	Branch the run commits to.
<code>--max-hours <n></code>	Wall-clock limit for the whole run.
<code>--push</code> , <code>--limit <n></code>	Used for overnight port batches. Exact meaning [TO CONFIRM].
<code>--accept-respec <name></code>	Publish a recapture of a ported component when two named hashes match.

Configuration values

Setting	Value
Per-capture time limit	20 minutes by default. <code>timeoutMs</code> 2400000 (40 min) for usage-meter, time-picker, swipe-actions.
Navigation retry ladder	5s, 15s, 45s, for HTTP 5xx, 429 and transient <code>net::ERR_*</code> .
Port attempts per component	3. Port breaker at 4. Preflight breaker off.
Animation drain cap in checks	5s.
Token tolerance	A value more than 2px from a token needs a decision in <code>tokens-overrides.json</code> .

Setting	Value
Capture-config keys	<code>openSelector</code> , <code>overlaySelector</code> , <code>rootSelector</code> , <code>dismissOnPress</code> , <code>timeoutMs</code> . Example: <code>breadcrumb rootSelector</code> is <code>#preview-canvas nav ol[class*="breadcrumb_list__"]</code> .
Bottom-sheet override	<code>calc(var(--sheet-extension) + env(safe-area-inset-bottom, 0px))</code> pinned to <code>sheet padding-bottom</code> , with <code>--sheet-extension</code> : <code>160px</code> .
text-reveal <code>.clip</code> overrides	<code>.1em</code> only on padding-top, <code>.16em</code> only on the sides, <code>.24em</code> only on padding-bottom, negatives only on margins.
Typical port cost	About \$0.55 to \$1.75 per component (API-equivalent), about \$1.09 for a simple one.

Token scales

Spacing (CSS variables look like `--space-related`):

Token	none	hair	tight	related	group	inset	comfort	section	block	page	stage	screen
px	0	2	4	8	12	16	20	24	32	48	64	80

Radius (CSS variables look like `--radius-container`):

Token	none	tight	chip	control	container	badge	overlay	full
px	0	2	4	6	8	11	12	999

11. Open items and future work

- ☐ **Confirm the text-reveal re-port.** It was started after the final recapture. Check `driver-state.json` and `REPORT.md` for its status.
- ☐ **Review by eye in Storybook:** breadcrumb (enter and exit animation, reduced motion), search-field (clear button), text-reveal (stagger).

- ☐ **Wider port trial** on 6 to 8 harder components covering an overlay (dialog or tooltip), a chart, a form field (select or combobox) and a motion-heavy one (carousel). The first ports were all simple.
- ☐ **Decide on merging** `port-trial-1` and `port-trial-2` into `react-migration`. Nothing is merged.
- ☐ **Cloud parity canary**, then overnight port batches with `--resume`, expecting Claude usage limits.
- ☐ **Close the inheritance-check gap**: add a `directParent` field, recapture the covered specs, then wire the check as a gate.
- ☐ **Distribution decision**: npm package or copy-in registry.
- ☐ **Per-component descriptor file** so screen-design agents can use components without reading code.
- ☐ **Refresh the stale Docs project document** about this work.
- ☐ **Breadcrumb product decision**: accept the plain `<a>` instead of `next/link`, or add an adapter.
- ☐ **Cosmetic**: record button and tabs as `done` rather than `preflighted`.
- ☐ **Idea, not started**: render Arc's own component source locally instead of capturing uiarc.dev. It would be faster and could run in the cloud. Open questions: does `reference/arc-library` include the global CSS, theme variables and fonts, is a demo harness needed per component, and does a local render match the live site. Proposed first step is a read-only feasibility study on 3 to 5 components.

Gaps to fill

These were not stated in the project history, so they are marked [TO CONFIRM] above:

1. Dependency install steps (is `npm install` enough, which Node version).
2. The exact way to set a component to `needs-recapture` (script, flag or manual edit of `driver-state.json`).
3. The exact flag names for `--accept-respec` hashes, and the meaning of `--limit` and `--push`.
4. Commands for the driver selftest, node tests and capture tests.
5. How the Next.js consumer at localhost:3007 is started.
6. Locations of `tokens-map.json` and `deviations.json`, and the path of Arc's CSS modules inside `reference/arc-library`.

7. The full list of token CSS variable names (only `--space-related` and `--radius-container` were seen).
8. The outcome of the text-reveal re-port, and whether newer commits exist after `6d174abc` .
9. The current count of components by status after that re-port.
10. Who approves deviations and token overrides besides Anil Choudhary, and the review process before merging.
11. Any deadline or launch date for Silicon UI.