PROMPT — Convert Angular Components to Existing Libraries (Token■optimized)

Goal:

Convert an existing component into an already created Angular library. Do not restructure the monorepo. Every target lib must depend on `@org/core` (global Tailwind + theme).

Input (required each run):

source_path

target_lib (@org/)

component_name

external_dependencies (optional)

Rules:

- Convert component only. No new libraries.
- Tailwind and theme must come from `@org/core` only.
- SCSS must use @use tokens/mixins from core.
- Export component in lib public-api.ts.
- No changes to angular.json, tsconfig.*, tailwind config.
- Keep API behavior same. No breaking changes.
- Add minimal a11y (roles/aria when applicable).
- Add basic spec test rendering + main inputs/outputs.
- If missing, add @org/core to package.json (workspace:*).

Core usage examples:

SCSS:

@use '@org/core/styles/tokens' as tokens;

@use '@org/core/styles/mixins' as mixins;

TS utils:

import { DomUtils } from '@org/core';

Output (always):

- Component TS, HTML, SCSS, spec.ts
- Update public-api.ts export
- Update package.json dependency if needed
- Short migration notes (inputs/outputs mapped, design tokens applied)

Success criteria:

- Library builds clean
- Component uses @org/core theme/Tailwind
- Same visual style as core design system
- Tests pass

Execution example:

[CONVERT]

source_path: apps/legacy/button/

target_lib: @org/buttons

component_name: OrgButton

external_dependencies: none

Produce final component code + exports + migration notes.