

Painting Plone Live

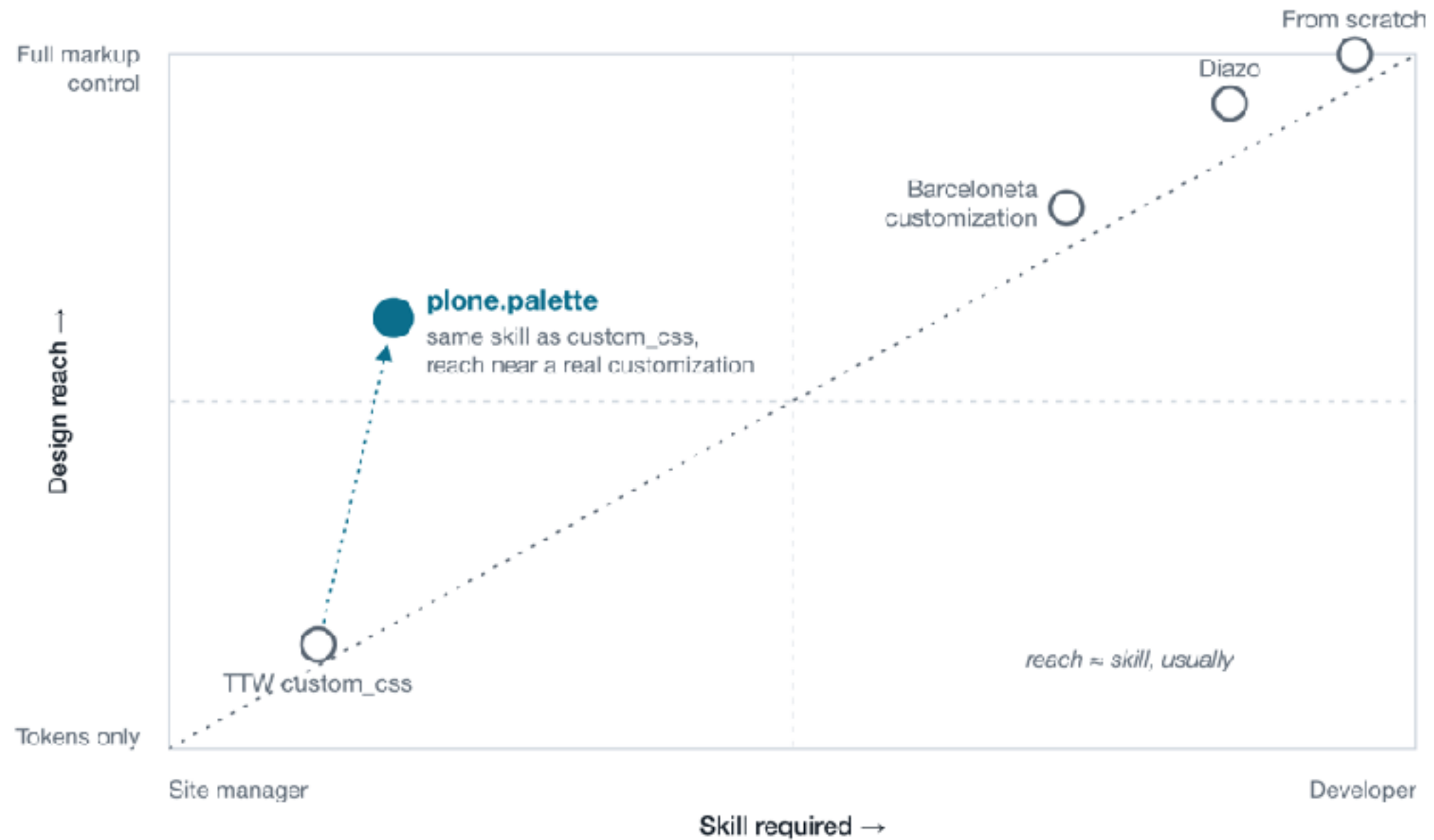
Runtime Bootstrap theming for Plone Blicca with plone.palette
Roland Fasching · Plone Conference 2026



Four ways to theme Classic UI

- Custom Styles TTW in custom_css field — free text, no build, filled up with '!important rules'
- Barceloneta SCSS — compile, restart
- Diazo — rewrites markup, compile
- From scratch — everything
- plone.palette: the same variables — no build step

Where plone.palette sits



Why this is possible now

- Bootstrap 4 (2018): variables exported, but unused
- Bootstrap 5.2 (2022): components read their own variables
- Bootstrap 5.3 (2023): color modes, semantic variables
- Bootstrap 6: same idea, new names

The solution — plone.palette

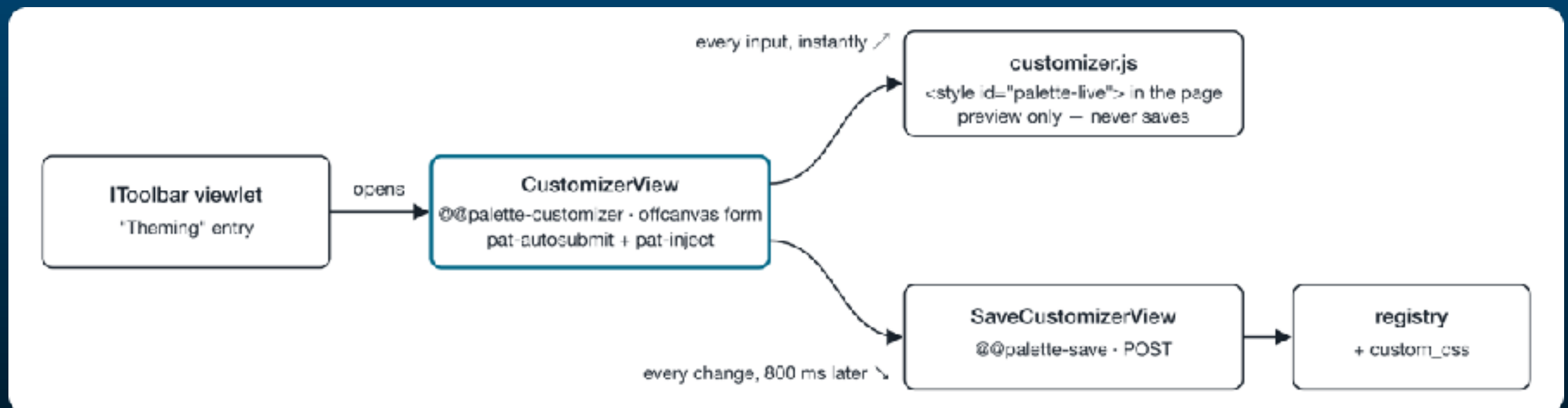
- Toolbar → Theming → offcanvas panel
- Live preview on the real page
- Every change is saved — registry + custom_css
- No save button, no SCSS, no restart

What it's for — and what it isn't

- For: brand changes by a site manager
- For: any Bootstrap CSS variable
- For: one theme package, many branded sites
- For: Prototyping
- Not: layout or markup (Diazo, mosaic)
- Not: pixel-perfect custom design

Live demo

How it's built — four components



Why does it work - the cascade trick

- Winning without !important
- :root { --bs-primary: ... } — loads last, wins
- .btn-primary { --bs-btn-bg: ... } — re-declared per component, so do we
- Same for RGB variants, links, outline buttons

JS driven by data attributes

- data-css-var + data-css-unit → :root { var: value }
- data-css-selector + data-css-prop → selector { prop: value }
- New field: one Python table row — no JavaScript, no template

Ready for Bootstrap 6

- `--bs-primary` ↔ `--primary-base`
- Every variable is written under both names
- A test checks that both lists match
- Blicca with Bootstrap 6: experimental — not installed, not demoed today

Google Fonts, two ways

- With an API key: the full catalogue, cached one hour
- Without: about 80 popular fonts built in
- Live: a `<link>` tag — saved: an `@import` at the top of the CSS

Reset and uninstall

- Reset: every field to default, stylesheet cleared — one step
- Uninstall: removes palette's registry records and JS bundle
- custom_css stays — it belongs to plone.app.theming

An agent as customizer client

- `generate_css(records) -> str`
- No request, no session, no browser
- Anything that writes the registry can theme the site
- Human in a panel, agent in a script — same code

What to take home

- Colours, fonts, radius — no developer needed
- Build your theme on CSS variables — then it is live themable too
- Theming becomes data — a form, a script, an agent
- Try the beta: github.com/collective/plone.palette

Thank you

github.com/collective/plone.palette
Questions?

