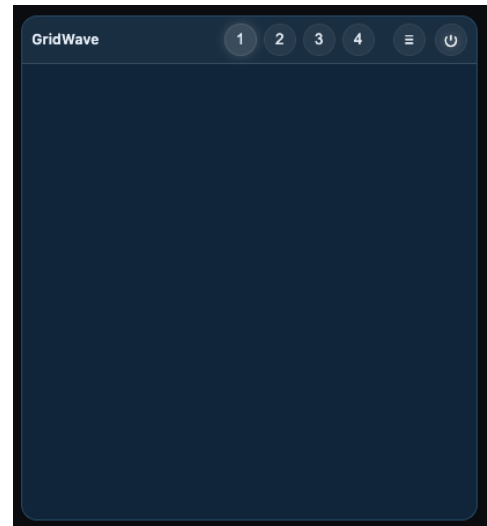


UID-Explore · Widgets 1.4

Support Layer · Widget Logic

Stand 03.10.2025 · Version 1.4.3



1) Architektur im Überblick

Ziel: Der *Support Layer · Widget Logic* stellt generische Bausteine für UI-Header und Aktionen bereit, die von Präsentations- und Input-Tools wiederverwendet werden. Alle Bausteine sind **ESM-Module**, interagieren **lose gekoppelt** über den **Event-Bus** (EBUS) und spiegeln ausgewählte Zustände im DOM (Attribute/Events).

One-Header-Policy: Es gibt pro Widget genau **einen** Header, den Tools *verwenden*, aber **nicht** selbst erzeugen.

Schichten (kontextualisiert)

- **Input (12-1_input):** sendet Parameter-Events (z. B. `uid:e:params:change`).
- **Base (12-2_base):** Engine/Director; berechnet Zeitreihen und publiziert `uid:e:model:update`, `uid:e:sim:data`.
- **Presentation (12-3_presentation):** visualisiert Serien (Chart, KPI, Grid Wave etc.).
- **Support (12-4_support):** diese Widget-Logik (Header, Actions, Dropdown/Burger, Off-Policy, Styles, Tooltips).
- **Bus:** `on/getLast/emit` + DOM-Spiegelung (CustomEvents) — identische Konventionen wie in den Visual-Modulen.

UID Explore · Grid Wave

2) Dateien & Rollen (Quick-Map)

Kern-Widgets

- `/uid/12-4_support/widgets/index.js` – Public-API + Boot (Styles/Hub/Tooltip).
- `/uid/12-4_support/widgets/header.js` – Header erzeugen/sichern (Titel, Dyn/Globals, Burger, Power, ARIA-Sync).
- `/uid/12-4_support/widgets/actions.js` – Actions in Header-Slots mounten (dyn/globals) inkl. Burger-Modell-Bridge.
- `/uid/12-4_support/widgets/registry.js` – Preset-Fabriken (segmented/toggle/button/dropdown/multitoggle/scaleToggle).
- `/uid/12-4_support/widgets/contracts.js` – *validateSpec* (leichte Action-Schema-Prüfung).

- `/uid/12-4_support/widgets/hub.js` – Overlay-Hub (open/close, openMenu).
Signatur: `open(anchorBtn, menuEl)`.
- `/uid/12-4_support/widgets/header.tools.js` – Globale Header-Tools (Scale %/#, HIT).
- `/uid/12-4_support/widgets/off-policy.js` – Sichtbarkeit im OFF (Power-Gating + Chart-Mini-Mode).
- `/uid/12-4_support/widgets/rehydrate.js` – Replays letzter Bus-Events + Auto-Rehydrate bei Power-ON.
- `/uid/12-4_support/widgets/measure.js` – Header-Collapse messen (ResizeObserver + rAF-Debounce).
- `/uid/12-4_support/widgets/styles.js` – Einmalige Style-Injection (Tokens, Buttons, Dropdown).
- `/uid/12-4_support/widgets/tooltip.js` – Globaler Cursor-Tooltip (data-wa-tip/title; clamp, ESC).
- `/uid/12-4_support/widgets/tips.js` – Fokus/Hover-Tooltip-Layer (role=tooltip) — **nicht mit tooltip.js kombinieren.**

Renderer (`/uid/12-4_support/widgets/render/`)

- `button.js` – `mkBtn()` + `renderButton()`; runde Header-Buttons per Inline-Style, A11y für Icon-only.
- `segmented.js` – exklusive Gruppe (role=radiogroup) mit Tastatur-Nav (←/→, Home/End).
- `toggle.js` – additive Toggles im Segmented-Look, gruppiert als `.wa-seg.wa-multi` (role=group).
- `multitoggle.js` – additive Segment-Toggles (1..n), `onChange(value, isOn)`.
- `dropdown.js` – Dropdown/Burger-Panel (Simple-List oder Sections-Grid), Hub-gesteuert.

3) Datenfluss & Boot

1. **Import:** `widgets/index.js` sorgt einmalig pro Seite für Styles (`ensureWidgetStyles()`), lädt Hub/Tooltip (Side-Effects) und exportiert Public-APIs.
2. **Header:** `attachWidgetHeader(host, {...})` erzeugt (falls nötig) den Header und platziert Slots:
 - **Dyn** (links/rechts ausrichtbar): primäre Tool-Wechsler (Segmented, Additive).
 - **Globals** (rechts): globale Controls (Scale/HIT, Dropdown).
 - **Burger:** öffnet Menü (Sections-Grid).
 - **Power:** toggelt OFF/ON (siehe Off-Policy).
3. **Actions:** `mountWidgetActions(host, spec)` rendert Controls in die Slots. Das Widget kann ein **Burger-Modell** bereitstellen (explizit oder heuristisch aus `globals`).
4. **Off/On:** `applyOffState()` setzt `data-enabled` + ggf. Chart-Mini-Mode und feuert bei ON `uid:widget:power:on` (Auto-Rehydrate-Hook).
5. **Rehydrate:** `attachAutoRehydrate(host, bus, events)` reagiert auf ON-Signale sowie Attributwechsel und **replayt** letzte Bus-Events (`getLast-emit`) — robust gegenüber Lazy-Mounts.

4) Interaktion (UI/UX)

- **Segmented (exklusiv):** 1-of-N, radiogroup, Maus + Tastatur.
- **Toggle / MultiToggle (additiv):** mehrere parallele Zustände, `aria-pressed + .is-active`.
- **Dropdown/Burger:** `aria-haspopup="menu"`, Items als `menuitem`, `menuitemradio`, `menuitemcheckbox`; ✓-Markierung und Gruppen-Exklusivität über Rolle/ARIA.
- **Globals · Scale/HIT:**
 - `%/#` skaliert Darstellung per `uid:e:viz:scale:set + uid:e:viz:scale:changed`.
 - **HIT** (Herdimunität) toggelt `uid:e:lines:hit:toggle` (SIS deaktiviert).
- **Power:** schaltet Widget-UI aus/ein; bei ON werden Rehydrate-Events erneut emittiert → UI konsistent.

5) Event-Referenz (Support-Sicht)

Konsumiert

- `uid:widget:power:on` (DOM, CustomEvent) → *Auto-Rehydrate* starten.
- `uid:e:viz:scale:changed {mode:'pct' | 'abs', scope?}` → Sync des `%/#`-Buttons.
- `uid:e:lines:hit:toggle {on:boolean, scope?}` → Sync des **HIT**-Buttons.

Emittiert (typisch)

- `%/#`:
 - `uid:e:viz:scale:set {mode:'pct' | 'abs', scope}`
 - `uid:e:viz:scale:changed {mode:'pct' | 'abs', scope}`
- **HIT:** `uid:e:lines:hit:toggle {on, scope}` (+ optionales `uid:e:kpi:pulse {kind:'HIT'}`).
- **Tool-Primärfeld** (Template in `actions.js`): `${NS}:${SEG_KIND} { [SEG_KIND]: <name> } und optional { value:'1'.. } mit source:'wa'.`
- **Seeds/Replay** (optional, Template): `uid:e:params:change, uid:e:data:series, uid:e:sim:*` (nur wenn `FEATURES.seeds=true` im Actions-Template).

Konventionen

- **source: 'wa'** wird vom Actions-Template automatisch ergänzt (Transparenz der Auslöser).
- **DOM-Spiegelung:** System-Events sind als `window-CustomEvents` observierbar (Debug).

6) Accessibility (A11y) & I18N

- **Segmented:** Container `role="radiogroup"`, Buttons `role="radio"`, `aria-checked`, `tabindex`-Verwaltung, Keyboard (`←/→/Home/End`).
 - **Toggle/MultiToggle:** `aria-pressed=true|false`, Wrapper `role="group"` mit optionalem `aria-label`.
 - **Dropdown/Burger:** Button `aria-haspopup="menu"`, `aria-expanded`; Items rollengetreu, ✓-Status via `aria-checked`.
 - **I18N:** einfache DE/EN-Heuristik per `document.documentElement.lang`, z. B. Tooltips/Title „Optionen/Options“, „Ein/Aus/On/Off“.
 - **Focus:** Hub gibt Fokus an Anker zurück (ESC), Tooltip reagiert auf ESC/Resize/Scroll.
-

7) Styles & Theming

- **Injection:** `ensureWidgetStyles()` fügt `style#uid-widgets-v1` genau einmal ein.
 - **Tokens** (`:root`): `--wa-fg`, `--wa-bg`, `--wa-border`, `--wa-active`, `--wa-radius`, `--wa-size` usw.
 - **Header-Buttons:** Kreisform über Inline-Styles in `render/button.js` (übersteuert Pill-Look).
 - **Dropdown:** `.wa-menu` **ohne** `position:fixed`; Positionierung übernimmt der Hub-Host (absolut) für verlässliche Anker-Platzierung.
 - **Collapse-Attribute:** `data-collapse-icon-only`, `data-collapse-actions` (wird von `measure.js` gesetzt).
-

8) Performance & Robustheit

- **Measure:** `ResizeObserver` + **rAF-Debounce** → kein Thrash bei Layout-Shifts.
 - **Hub:** schließt auf `scroll/resize/orientationchange` und `Outside-Click`; räumt Listener beim `close()` sauber ab.
 - **Renderer:** geben Disposer zurück (`remove-Listener/DOM`), Cleanup von Wrappern wenn leer.
 - **Tooltips:** leichtgewichtig; `tooltip.js` (Cursor-Tooltip) und `tips.js` (Hover/Fokus-Layer) **nicht gleichzeitig** aktivieren, um Doppelanzeigen zu vermeiden.
 - **Speicher/Leaks:** durchgängig `unsubscribe/dispose`-Muster; Events defensiv mit `try{}` umschlossen.
-

9) Verträge & Modelle

9.1 Actions-Contract (*leichtgewichtiges Shape-Checking*)

`validateSpec(spec, {debug})` prüft: `spec.dyn / spec.globals` (Arrays), Action-Typen (segmented|toggle|button|dropdown|multitoggle) und Minimalfelder (z. B. `segmented.options, toggle.get, dropdown.items`). Warnungen bei Debug.

9.2 Presets (Fabriken) – Beispiele

- `segmented({id, options:[{label,value,title?}], get, set})`
- `toggle({id,label?,icon?,get,set})`
- `button({id,label?,icon?,onClick})`
- `dropdown({id,label?,items|sections})`
- `multitoggle({...})` (generisch)
- `scaleToggle({get,set})` (bequem: `%↔#`)

9.3 Burger-Modell (Sections-Grid)

Explizit über `spec.burgerSections` **oder** heuristisch aus `globals` (Toggles/Dropdowns). Hub erstellt Menü-DOM via `openMenu(anchor, model)`; Gridlayout in `render/dropdown.js` analog.

10) Off-Policy & Power-Gating

- **Standard:** Bei OFF (`data-enabled="false"`) blendet der Header **Actions/Burger** aus; Power bleibt sichtbar.
- **Spezialfall Chart:** OFF → `data-collapse="mini"` (historisch kompatibel auch `data-off-mode="mini"`); nur Basisanzeige.
- **ON-Signal:** Beim Einschalten dispatcht die Policy `uid:widget:power:on` (bubbles) → *Auto-Rehydrate*.

11) Rehydrate (Replay-Flow)

- **`rehydrateWidget(bus, events)`:** Für jede `{ev, fallback?}` wird `getLast(ev)` geprüft; wenn vorhanden, erneut `emit(ev, last)`. Falls **kein** Last-Wert und `fallback` definiert, wird **einmalig** der Fallback emittiert.
- **`attachAutoRehydrate(host, bus, events, header?)`:** Hört auf
 - `data-enabled` am Header,
 - `data-widget-enabled` am Host,
 - `uid:widget:power:on` (CustomEvent).→ In allen Fällen: `rehydrateWidget(...)`.
- **Defaults:** `DEFAULT_EXPLORE_EVENTS` enthält typische Explore-Themen (`model:params, data:series, sim:status/pointer, viz:scale:changed`).

12) Hub (Overlay-Host)

- **API:**
 - `open(anchorBtn, menuEl)` – *Signatur ist wichtig (Anker zuerst)*.
 - `close()` – Event-Listener/DOM entfernen, `aria-expanded=false`.
 - `openMenu(anchorBtn, model, {columns})` – Sections-Grid bauen und über `open()` anzeigen.
- **A11y:** Host `role="menu"`, ESC schließt, Fokus wird nach Schließen an Anker zurückgegeben.

Hinweis: In `actions.js` wurde der Burger-Aufruf konsolidiert: `open(anchorBtn, menu)` (korrigierte Reihenfolge).

13) QA-/Abnahme-Checkliste

1. **Header-Integrität:** Genau ein `.uidw-header`; Slots vorhanden (`.uidw-actions-dyn`, `.uidw-actions-globals`).
 2. **Burger:** Menü öffnet neben Anker; Radio-Gruppen schalten exklusiv; ESC/Outside-Click schließt.
 3. **Power/Off-Policy:** OFF blendet Actions aus; ON triggert `uid:widget:power:on`; *Auto-Rehydrate* feuert.
 4. **Segmented:** radiogroup-Rollen korrekt; Tastatur-Nav (`←/→/Home/End`) funktioniert; aktueller Button `tabindex=0`.
 5. **Toggle/MultiToggle:** `aria-pressed` spiegelt Status; Gruppenlabel (falls konfiguriert) vorhanden.
 6. **Dropdown:** `menuitem(radio|checkbox)` trägt `aria-checked`; ✓-Symbol nur bei ausgewählten Items.
 7. **Measure:** Bei schmalem Platz schaltet `data-collapse-icon-only/actions`; Burger erscheint.
 8. **Styles:** Stylesheet wird genau einmal injiziert; `.wa-menu` positioniert korrekt (ohne `position:fixed`).
 9. **Tooltips:** Präzise auf Cursor/Element; **nur eines** der Systeme aktiv (global `tooltip.js` *oder* fokusbasiert `tips.js`).
 10. **Dispose:** Alle Renderer geben Disposer zurück; Abmeldung/DOM-Entfernung klappt ohne Fehler.
-

14) Typische Fehlerbilder & Gegenmittel

- **Falsche Hub-Signatur** → Menü erscheint an falscher Stelle. **Fix:** `Hub.open(anchorBtn, menuEl)`.
- **Doppelte Tooltips** → `tooltip.js` und `tips.js` parallel aktiv. **Fix:** eines deaktivieren.
- **ARIA stale nach Re-Render** → Header-ARIA-Sync (`header.js`) greift erst nach Mutations. **Fix:** auf `.uidw-actions-dyn` hören (MO) — bereits implementiert.
- **Leere Wrapper** nach Dispose → Renderer reinigen Wrapper nur, wenn keine `.wa-seg-btn` mehr vorhanden — verhindert Flackern.

- **Scroll schließt Menü nicht** → Hub-Listener auf `scroll/resize/orientationchange` prüfen — implementiert.
-

15) Mini-Rezepte (Copy/Paste)

Dropdown von außen synchronisieren

```
// Auswahl für Gruppe 'fmt' auf 'percent' setzen:  
dispose.sync({ fmt: 'percent' });
```

Power-ON triggern (z. B. nach externem Toggle)

```
headerEl.dispatchEvent(new CustomEvent('uid:widget:power:on', { bubbles:  
true }));
```

Scale von außen setzen

```
EBUS.emit('uid:e:viz:scale:set', { mode: 'pct', scope: widgetEl });  
EBUS.emit('uid:e:viz:scale:changed', { mode: 'pct', scope: widgetEl });
```

16) Erweiterungsvorschläge

- **Legend-Sync** als Preset in `registry.js` (z. B. Sichtbarkeit S/E/I/R/V/D via Dropdown-Checkboxes).
 - **Persistenz-Keys** für Segmented/Groups standardisieren (Template bietet `PERSIST(root)` bereits).
 - **JSDoc-Typen**: Action-Shapes/Events annotieren (IDE-Intellisense).
 - **Snapshot-Tests**: UI-Zustände der Header-Controls (Segmented/Dropdown) per DOM-Abbild vergleichen.
-

17) Changelog (Dok) · v1.4.3

- **Hub.open-Signatur** im Burger-Pfad explizit dokumentiert und in `actions.js` angewandt.
 - **A11y-Hinweise** konsolidiert (Roles/ARIA-Attribute — Segmented/Dropdown).
 - **QA-Checkliste** und Fehlerbilder aktualisiert.
-

Anhang A — Action-Shapes (Kurzreferenz)

Segmented

```
{ type:'segmented', id, options:[{label?, value, title?, icon?,  
disabled?}], get():string, set(v), onChange?(v) }
```

Toggle

```
{ type:'toggle', id, label?, icon?, get():boolean, set(on:boolean),  
onToggle?(on), onChange?(on) }
```

Dropdown (Simple-List)

```
{ type:'dropdown', id, label?, items:[{label, icon?, onSelect?, selected?,  
isSelected?()}], autoClose?:boolean }
```

Dropdown (Sections-Grid)

```
{ type:'dropdown', id, label?, sections:[{ title?, items:[{label, value?,  
type:'rad
```