**07 Size Concept**

# **Size Concept**

**Date:** September 17, 2025  
**Version:** v1.0 - Extracted from combined density-size concept  
**Status:** Done in https://github.com/oblique-bit/oblique/tree/tokens-develop-02.05

## **Overview**

This document outlines the **Size** system for component scaling through unified size modes, ensuring consistent component relationships and accessible user experiences.

## **Design goals**

### **\*\*Harmonious component integration\*\***

* Components work together consistently without visual conflicts
* Size relationships feel natural and intentional
* Proper component hierarchy maintained across all size variants

### **\*\*Effortless user experience\*\***

* Consistent interaction patterns across different contexts
* Predictable sizing behavior reduces cognitive load
* Context-appropriate scaling for different user types

### **\*\*Accessible by design\*\***

* All size variants maintain usability standards
* Touch targets and readability preserved across scales

### **\*\*Developer experience\*\***

* Components automatically coordinate without requiring manual size matching by developers
* Clear inheritance patterns reduce implementation complexity
* Consistent API across size system

---

## **Size system (done)**

### **\*\*Definition\*\***

* \*\*Triggers inset spacing\*\* (inside components, mostly atoms and molecules)
* \*\*Determines size of components\*\* (primarily when grouped, in relation and context)

### **\*\*Implementation\*\***

* \*\*Figma\*\*: Variable collection with modes `sm`, `md`, `lg`
* \*\*Tokens Studio\*\*: Theme group with themes `sm`, `md`, `lg`
* \*\*Current naming\*\*: Uses `component-size` theme group in `$themes.json` (located at `/src/lib/themes/`) - called this way because only certain components react to it currently, but can be expanded and renamed later when more components support size modes

### **\*\*Shared component sizes philosophy\*\***

Shared sizes are beneficial for:

* \*\*Visual and interaction consistency\*\* (end user experience)
* \*\*Efficiency\*\* (product designer workflow)
* \*\*All components share one size collection\*\*
* \*\*Contextual inheritance\*\* (sizing flows through interface hierarchy)
* \*\*Component token boundaries\*\* (each component defines its supported size range)
* \*\*Reduced cognitive load\*\* (designers work with unified size modes rather than individual component sizing decisions)

### **\*\*Individual visual size with unified naming\*\***

* \*\*Universal naming\*\*: `sm`, `md`, `lg` across all components
* \*\*Component-relative scaling\*\*: Each component's "large" is large relative to itself
* \*\*Visual compatibility\*\*: LG button is larger than LG badge, but both are "large" and visually harmonious
* \*\*Maintains relationships\*\*: Badge can't be larger than a button at the same size level

### **\*\*What one size represents (e.g., LG)\*\***

* \*\*LG version of component itself\*\*: Each component scales relative to its own baseline
* \*\*Context for grouped components\*\*: All components react and follow the size context/mode
* \*\*Harmonious relationships\*\*: Components maintain proportional relationships across sizes

### **\*\*Shared size context with boundaries\*\***

Size functions as an overarching contextual layer that flows through the interface hierarchy:

**Global size declaration**: Container sets size context (sm/md/lg) for all descendants  
.form-section[data-size="lg"] {  
--size-context: lg;  
}

**Component boundary respect**: Each component inherits the global context. Individual context can be set like for the navigation, both in Figma (baked-in variable mode inside the component) and code.

**Design system governance**: Maintainers define and document size boundaries based on:

* \*\*Functional constraints\*\*: Minimum usability requirements
* \*\*Visual hierarchy\*\*: Maintaining component relationships
* \*\*Context appropriateness\*\*: Preventing visual overwhelm or under-emphasis

### **\*\*Proportional constraints\*\***

Some components require **width = height** to maintain visual proportion:

**Square components:**

* \*\*icon\_holder\*\*: Must maintain 1:1 aspect ratio for visual harmony

**Token structure:**  
"ob.size.icon-holder.dimension": {  
"sm": "16px", // height = width = 16px  
"md": "20px", // height = width = 20px  
"lg": "24px" // height = width = 24px  
}

---

## **Spacing roles**

### **\*\*Size system spacing\*\***

* \*\*Spacing as tool\*\*: Controls internal padding, gaps within components
* \*\*Affects\*\*: Button padding, input padding, icon gaps, internal component spacing
* \*\*Does not affect\*\*: Layout spacing between different components

---

## **Context awareness and inheritance**

### **\*\*Component role classification\*\***

#### **\*\*Size Context Providers\*\* (Set sizing context)**

* \*\*`dialog`\*\* - Sets density context for all contents
* \*\*`form`\*\* - Sets height-matching context
* \*\*`input`\*\* - Sets context for tags, buttons, badges - also if no overarching context set by form
* \*\*`service-nav`\*\* - Sets compact context for actions

#### **\*\*Size Context Consumers\*\* (Inherit from context)**

* \*\*`tag`\*\* - Inherits from input fields
* \*\*`button`\*\* - Inherits in form/dialog contexts
* \*\*`badge`\*\* - Inherits from nearby components
* \*\*`icon`\*\* - Always inherits from parent

### **\*\*Size coordination principle\*\***

* \*\*Input text field LG ≠ badge LG\*\* (different visual sizes)
* \*\*Both maintain visual relation\*\* when scaled together
* \*\*Container-level inheritance\*\*: Set MD on container → all children inherit MD and scale together
* \*\*Dimensional relationships preserved\*\* across size changes

### **\*\*Inheritance types and key relationships\*\***

#### **\*\*Inheritance Patterns\*\***

1. \*\*Locked inheritance from parents\*\* (automatic mode/token inheritance pro component)
2. \*\*Overridable inheritance\*\* (can be changed if needed)
3. \*\*No inheritance\*\* (totally free - MD as default, but designer can override via variable modes)

#### **\*\*Key Component Relationships\*\***

* \*\*Input + Tag inside\*\*: Height matching for multi-select scenarios
* \*\*Input + Button\*\*: Form row alignment
* \*\*Dialog + All Contents\*\*: Density inheritance
* \*\*Infobox + Button\*\*: Action buttons in notifications
* \*\*Any Component + Tooltip\*\*: Context-aware help text
* \*\*Any Component + Icon\*\*: Always inherits from parent component

### **\*\*Scaling Direction\*\***

* \*\*Primarily vertical scaling\*\* (up and down)
* \*\*Horizontal scaling\*\* as secondary consideration for limited set of organisms such as Dialog.

---

## **Component sizing classification**

### **\*\*Size boundary governance\*\***

Components are classified by their supported size range, determined by design system maintainers based on functional and visual constraints.

### **\*\*Full size range: sm, md, lg\*\***

* \*\*Input components\*\*: text field, text area, checkbox, radio
* \*\*Interactive elements\*\*: badge, pill, button
* \*\*Rationale\*\*: High contextual flexibility needed for different user scenarios

### **\*\*Limited size range: md, lg\*\***

* \*\*Information components\*\*: infobox, alert
* \*\*Rationale\*\*: Small sizes compromise readability and information hierarchy

### **\*\*Minimal size range: sm, md\*\***

* \*\*Contextual components\*\*: tooltip, popover
* \*\*Rationale\*\*: Large sizes become visually overwhelming and break spatial relationships

### **\*\*Width-only sizing\*\***

* \*\*Modal components\*\*: dialog, modal
* \*\*Rationale\*\*: Height scaling handled by content, width responds to size context
* \*\*HowTo:\*\*: Only width-property consumes token switchable by component-size-modes and reacts on mode/context.

### **\*\*Fixed size (static size)\*\***

* \*\*Layout components\*\*: master-layout navigation, paginator
* \*\*Rationale\*\*: Consistency and recognition prioritized over size flexibility
* \*\*HowTo:\*\*: Consumes static size/dimension tokens that do not react on mode/context.

---

**Core principle**: Size acts as an overarching context mode (sm/md/lg) that flows through the interface, but individual components have predetermined size boundaries set by design system maintainers.

**Technical implementation**:

* Components respect global size context when within their supported range
* Components fallback to nearest boundary when context exceeds their limits
* Size boundaries are documented and enforced at token level

### **\*\*Size inheritance workflow\*\***

* \*\*Large group level\*\*: Apply to organisms, sections, templates
* \*\*Contextual overrides\*\*: Product designers can override for specific elements (e.g., primary conversion buttons)
* \*\*Business/UX exceptions\*\*: Requirements can override grouped size patterns

### **\*\*Development approach\*\***

* \*\*Holistic unification\*\*: Size system implemented in single pass when all components ready
* \*\*Workflow\*\*: Set LG container → all inside components inherit LG

### **\*\*Accessibility implementation\*\***

* \*\*Touch target maintenance\*\*: Minimum 44px touch targets preserved across all size variants
* \*\*Readability standards\*\*: Text contrast and sizing meet WCAG guidelines
* \*\*Keyboard navigation\*\*: Focus indicators properly sized for all contexts
* \*\*Screen reader compatibility\*\*: Size changes don't break assistive technology

---

## **Size and density interplay**

### **\*\*Coordinated combinations and design context matching\*\***

#### **\*\*Compact density + small size\*\***

* \*\*Use case\*\*: Frequently used apps for power user personas
* \*\*User type\*\*: Power users prioritizing information density
* \*\*Context\*\*: Data-heavy interfaces, administrative tools

#### **\*\*Medium density + medium size (default)\*\***

* \*\*Use case\*\*: Standard applications
* \*\*User type\*\*: Standard users requiring balanced approach
* \*\*Context\*\*: General purpose applications, dashboards

#### **\*\*Generous density + large size\*\***

* \*\*Use case\*\*: Simple flows, onboarding, simple dashboards, marketing websites
* \*\*User type\*\*: Users prioritizing clarity and ease of use
* \*\*Context\*\*: Marketing sites, onboarding flows, accessibility-focused interfaces

---

## **Key principles**

### **\*\*Visual hierarchy maintenance\*\***

* Size relationships ensure proper component hierarchy
* Larger components maintain visual dominance over smaller ones
* Consistency across contexts and use cases
* Components work together consistently without visual conflicts

### **\*\*Designer and developer efficiency\*\***

* Single size collection reduces decision fatigue
* Unified naming convention across all components
* Automatic inheritance reduces manual sizing work
* Clear implementation patterns for development teams

### **\*\*User experience consistency\*\***

* Predictable size relationships across interface
* Context-appropriate scaling for different user types
* Consistent interaction patterns across different contexts
* Maintained usability across all size variants

### **\*\*Accessibility First\*\***

* Touch target sizes maintained in coordinated components
* Text readability preserved in size modes
* Focus indicators properly sized for context
* Universal design principles applied to all size variants

---

## **Token examples**

### **\*\*Size token collection with boundaries\*\***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Token Name** | **Small** | **Medium** | **Large** | **Boundary** |
| `ob.size.input.height` | 32px | 40px | 48px | sm/md/lg |
| `ob.size.badge.height` | 20px | 24px | 28px | sm/md/lg |
| `ob.size.tag.height` | 24px | 32px | 40px | sm/md/lg |
| `ob.size.infobox.height` | 40px | 40px | 48px | md/lg only |
| `ob.size.tooltip.max-width` | 200px | 240px | 240px | sm/md only |
| `ob.size.navigation.height` | 56px | 56px | 56px | md fixed |
| `ob.size.icon.dimension` | 16px | 20px | 24px | sm/md/lg |

### **\*\*Boundary implementation tokens\*\***

|  |  |  |
| --- | --- | --- |
| **Token Name** | **Description** | **Implementation** |
| `ob.size.infobox.height.sm` | Falls back to md boundary | `{ob.size.infobox.height.md}` |
| `ob.size.tooltip.max-width.lg` | Falls back to md boundary | `{ob.size.tooltip.max-width.md}` |
| `ob.size.navigation.height.sm` | Ignores context, stays fixed | `{ob.size.navigation.height.md}` |
| `ob.size.navigation.height.lg` | Ignores context, stays fixed | `{ob.size.navigation.height.md}` |

### **\*\*Proportional square component tokens\*\***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Token Name** | **Small** | **Medium** | **Large** | **Constraint** |
| `ob.size.icon-holder.dimension` | 16px | 20px | 24px | width = height |

### **\*\*Three-dimensional token example (Viewport × Size × Component)\*\***

"ob.size.icon-holder.dimension": {  
"desktop": {  
"sm": "16px", // 16×16px square  
"md": "20px", // 20×20px square  
"lg": "24px" // 24×24px square  
},  
"mobile": {  
"sm": "20px", // 20×20px square (touch-friendly)  
"md": "24px", // 24×24px square  
"lg": "28px" // 28×28px square  
}  
}

### **\*\*Context-aware relationship tokens\*\***

|  |  |  |
| --- | --- | --- |
| **Token Name** | **Description** | **Value** |
| `ob.size.tag.input-coordinated.height.sm` | Tag height when in small input | `calc({ob.size.input.height.sm} - 8px)` |
| `ob.size.tag.input-coordinated.height.md` | Tag height when in medium input | `calc({ob.size.input.height.md} - 8px)` |
| `ob.size.tag.input-coordinated.height.lg` | Tag height when in large input | `calc({ob.size.input.height.lg} - 8px)` |

### **\*\*Combined token usage example\*\***

/\* Container sets size context \*/  
.form-section[data-size="sm"] {  
--size-context: sm;  
}

/\* Components inherit from context \*/  
.input {  
height: var(--ob-size-input-height-var(--size-context));  
}

.tag {  
height: var(--ob-size-tag-input-coordinated-height-var(--size-context));  
}

---

\*This document focuses specifically on the size system for component scaling and inset spacing control.\*