**Global Sizing Concept Research Questions**

# **Global Sizing Concept Research Questions**

**Date:** September 7, 2025  
**Purpose:** Define research questions for size-nesting concept based on competitive analysis of 46 design systems  
**Status:** Research Framework

## **\*\*Goal:\*\* Core Research Context**

### **Competitive Analysis Summary**

* \*\*46 Design Systems Analyzed\*\* via automated web scraping
* \*\*32 Systems Use Size Variants\*\*: `small/medium/large` pattern dominant (69%)
* \*\*41 Systems Have Sizing Components\*\*: Universal need for scalable components (89%)
* \*\*Common Sizing Components\*\*: button, input, tag, chip, badge, avatar, icon

### **Current Oblique Architecture**

* \*\*Size-Aware Components\*\*: Button (sm/md/lg), Badge (sm/lg), Pill (sm/md/lg), Tag (sm/md/lg)
* \*\*Token Structure\*\*: W3C DTCG compliant with CSS-aligned naming
* \*\*Semantic Scale\*\*: 24-point scale from `nano` to `supernova` with size mappings
* \*\*Implementation\*\*: Mode-based tokens with component-specific variations

---

## **\*\*Requirements:\*\* Research Questions Framework**

### **1. \*\*Size Inheritance Patterns\*\***

#### **Q1.1: Component Autonomy vs Inheritance**

* \*\*Question\*\*: Which components should have \*\*autonomous sizing\*\* (consumer-controlled) vs \*\*inherited sizing\*\* (parent-driven)?
* \*\*Investigation Focus\*\*:
* How do the 46 systems handle icon sizing inside buttons?
* When does a tag inherit input size vs maintain independent sizing?
* How do dismissible buttons inside tags relate to tag size?

#### **Q1.2: Nesting Hierarchy Rules**

* \*\*Question\*\*: What are the \*\*sizing hierarchy rules\*\* for nested components across design systems?
* \*\*Investigation Focus\*\*:
* Material Design's nested component behavior
* Carbon's compound component sizing patterns
* Chakra UI's size propagation through component trees

#### **Q1.3: Size Context Awareness**

* \*\*Question\*\*: How should components \*\*adapt to container constraints\*\* while respecting semantic sizing?
* \*\*Investigation Focus\*\*:
* How does a `lg` button behave inside a `sm` modal?
* Should chip size override when inside different input sizes?
* Container query vs component size token conflicts

### **2. \*\*Token Architecture Scalability\*\***

#### **Q2.1: Universal vs Component-Specific Tokens**

* \*\*Question\*\*: Should size tokens be \*\*universal\*\* (one scale for all) or \*\*component-specific\*\* (tailored per component)?
* \*\*Investigation Focus\*\*:
* How many systems use universal `sm/md/lg` vs component-specific sizing?
* Token maintenance overhead: universal vs specialized approaches
* Figma variable implementation complexity comparison

#### **Q2.2: Size Scale Granularity**

* \*\*Question\*\*: What is the \*\*optimal number of size steps\*\* for maximum utility without complexity?
* \*\*Investigation Focus\*\*:
* Distribution analysis: 2-size vs 3-size vs 4-size systems
* Most common patterns: `sm/md/lg` vs `xs/sm/md/lg/xl`
* Missing size needs in current 3-size (sm/md/lg) implementation

#### **Q2.3: Semantic Size Naming**

* \*\*Question\*\*: Should size naming be \*\*abstract\*\* (`sm/md/lg`) or \*\*semantic\*\* (`compact/spacious/hefty`)?
* \*\*Investigation Focus\*\*:
* Industry standard patterns across 46 systems
* Developer understanding and adoption rates
* Design handoff clarity: abstract vs semantic terms

### **3. \*\*Nesting Scenarios & Edge Cases\*\***

#### **Q3.1: Worst-Case Nesting Scenarios**

* \*\*Question\*\*: How should the system handle \*\*complex nested sizing relationships\*\*?
* \*\*Investigation Scenarios\*\*:

Input (lg) → Tag (md) → Dismiss Button (sm) → Icon (xs)  
Button (sm) → Icon (xs) + Text (xs) + Badge (xs)  
Modal (sm) → Form (md) → Input (lg) → Search Icon (md)

#### **Q3.2: Conflicting Size Requirements**

* \*\*Question\*\*: What happens when \*\*parent and child size requirements conflict\*\*?
* \*\*Investigation Focus\*\*:
* Override rules: parent wins vs child maintains autonomy
* User experience impact of size mismatches
* Accessibility considerations for nested sizing

#### **Q3.3: Responsive Size Behavior**

* \*\*Question\*\*: How should \*\*global sizing interact with responsive design\*\*?
* \*\*Investigation Focus\*\*:
* Viewport-based size adjustments vs fixed component sizing
* Container queries impact on nested component sizing
* Mobile-first sizing strategy for nested components

### **4. \*\*Developer Experience & Implementation\*\***

#### **Q4.1: Token Consumption Patterns**

* \*\*Question\*\*: How should developers \*\*consume size tokens\*\* in complex component hierarchies?
* \*\*Investigation Focus\*\*:
* API design: explicit sizing props vs automatic inheritance
* CSS custom property patterns for nested sizing
* TypeScript interface design for size relationships

#### **Q4.2: Figma-to-Code Translation**

* \*\*Question\*\*: How can \*\*Figma variable modes translate\*\* to nested component sizing in code?
* \*\*Investigation Focus\*\*:
* Variable collection structure for nested components
* Mode switching behavior for compound components
* Design token export for complex size relationships

#### **Q4.3: Maintenance & Consistency**

* \*\*Question\*\*: How can the system \*\*prevent size relationship bugs\*\* during development?
* \*\*Investigation Focus\*\*:
* Automated testing strategies for nested sizing
* Linting rules for size token usage
* Documentation patterns for size inheritance rules

---

## **\*\*Analysis:\*\* Investigation Methodology**

### **Phase 1: Competitive Pattern Analysis**

* \*\*Systems to Deep-Dive\*\*: Material Design, Carbon, Chakra UI, Ant Design, Adobe Spectrum
* \*\*Focus Areas\*\*: Component nesting documentation, size token architecture, developer APIs
* \*\*Documentation Review\*\*: Size inheritance rules, nesting guidelines, edge case handling

### **Phase 2: Current System Evaluation**

* \*\*Token Architecture Assessment\*\*: Universal Component-Sizes collection feasibility
* \*\*Component Mapping\*\*: Free vs locked component categorization
* \*\*Edge Case Testing\*\*: Complex nesting scenarios with current tokens

### **Phase 3: Proposal Development**

* \*\*Size Inheritance Rules\*\*: Clear hierarchy and override patterns
* \*\*Token Structure improvement\*\*: Balance between simplicity and flexibility
* \*\*Implementation Guidelines\*\*: Developer-friendly consumption patterns

---

## **\*\*Goal:\*\* Expected Outcomes**

### **Research Deliverables**

1. \*\*Component Classification Matrix\*\*: Free vs locked sizing by component type
2. \*\*Nesting Hierarchy Rules\*\*: Clear precedence and inheritance patterns
3. \*\*Token Architecture Proposal\*\*: improved structure based on competitive analysis
4. \*\*Edge Case Resolution Guide\*\*: Solutions for complex nested scenarios
5. \*\*Implementation Roadmap\*\*: Practical steps for global sizing concept

### **Success Criteria**

* \*\*Clear Decision Framework\*\*: When to use inherited vs autonomous sizing
* \*\*Scalable Token Architecture\*\*: Supports current and future component needs
* \*\*Developer-Friendly API\*\*: Simple consumption with predictable behavior
* \*\*Design System Alignment\*\*: Competitive analysis validation of approach

---

\*Research framework based on competitive analysis of 46 design systems\*  
\*Context: W3C DTCG compliant token architecture with CSS-aligned naming\*  
\*Goal: Tokenized, clear, scalable size-nesting concept\*