Modes Architecture I

```
semantic/
├── sizing.json # W3C DTCG compliant
├── spacing/desktop.json # Responsive multipliers
├── spacing/mobile.json # Responsive multipliers
├── color/s1-lightness/ # Light/dark themes
├── color/s2-emphasis/ # Emphasis modes
└── color/s3-semantic/ # Compiled semantic layer
global/themes-user/
├── lightness/ # Light/dark switching
└── viewport/ # Responsive system (4px/5px)
```

`

⚠️ Identified Gaps for VP

- `Missing: Unified` semantic/modes/ `architecture`
- `Missing:` component-size `theme configuration`
- `Missing: Basic` viewport `mode simplification (de p/mobile only)`
- `Inconsistent: Component size implementation patt`

Strategic Objectives

Primary Goal: Establish comprehensive modes architecture foundation with Phase 1 MVP implementation:

- **Big picture architecture** - Complete future vision with 11-dimensional modes system
- **Strategic roadmap** - Phased implementation strategy for systematic expansion
- **Architecture foundation** - Extensible framework for all future mode types
- **Phase 1 VP modes** - Essential foundation modes for immediate implementation

�🏗️ Complete Future Architecture Vision

📋 **Full odes Architecture Roadmap** *(All structures prepared, selective implementation)*

Phase 1: VP Foundation odes *(Primary Implementation Focus)*

- Contrast modes - standard/high (A11y color cont enhancement)

Phase 3: User Preference & Accessibility les

- otion modes - enabled/disabled/reduced (animati ccessibility controls)
- Typography modes - readable/standard/compact/A+/A/ yslexia-friendly (font weight, line height, letter spacing for readability + user preference typography scaling slexia support)
- Colorblind modes - standard/deuteranopia/protanopi anopia/achromatopsia (colorblind-friendly palettes for common color vision deficiencies)
- otion Preference modes - User-controlled animat settings with system integration
- Contrast Preference modes - User-controlled cont with system high-contrast support

Complete ulti-Dimensional Architecture

`

```
Full Future System (All 11 Mode Types):

lightness × interaction-emphasis × viewport × compone ize × density × contrast × motion × typography × user-font-scale × user-motion-pref × user-contrast-pref × dyslexia-su t

= 11-dimensional mode combinations
```

`

🎯 **VP Goal:** Prove modes architecture with essential ndation + targeted testing

VP odes Selection *(Phase 1 Essential I ementation)*

- ✅ **Lightness modes:** light/dark (existing theme l ness switching)
- ✅ **Interaction-emphasis modes:** low/standard/high isting component interaction states)
- 🔄 **Component Size modes:** sm/md/lg (WIP migrati component dimension coordination)
- 🆕 **Viewport modes:** desktop/mobile (basic 2-brea nt system for MVP)

VP Architecture: lightness × interaction-emphasis × com ent-size × viewport (4D system)
Note: Density modes moved to Phase 2 - focus on size e migration completion first

VP Component Testing Suite *(Strategic C onent Selection)*

🧪 Test Case 1: Simple Component - icon_holder

- **Purpose:** Baseline size mode behavior validation
- **odes tested:** All size modes (sm/md/lg)
- **Complexity:** Low (single dimension changes)

🧪 Test Case 2: Interactive Component - button

- **Purpose**: Interactive states + size mode coordinat
- **odes tested**: All size modes + interaction-emphas odes + lightness modes
- **Complexity**: Medium (multiple mode interactions)

🧪 Test Case 3: Content Component - tag

- **Purpose**: Text content + size mode relationships
- **odes tested**: All size modes + interaction-emphas odes + basic viewport modes
- **Complexity**: Medium (content-responsive sizing + v ort coordination)

🧪 Test Case 4: Status Component - infobox

- **Purpose**: Semantic color modes + worst-case status narios
- **odes tested**: All interaction-emphasis modes + li ess modes (all status colors)
- **Complexity**: High (color mode combinations + statu mantics)

🧪 Test Case 5: Composite Component - input text fiel

- **Purpose**: Babuschka doll complexity (input + tag + ove button inside)

- **Complexity**: Very High (nested component mode inhe ance + worst-case scenario)

VP Token Foundation *(Complete Architect e, Selective Population)*

```
`
semantic/
├── modes/ # 🆕 COMPLETE ARCHITECTURE PREPARED
| ├── component-size/ # ✅ MVP: Full implementation ( d/lg)
| | ├── sm.json
| | ├── md.json
| | └── lg.json
| ├── density/ # 🔄 STRUCTURE READY: Implementation d red from MVP
| | ├── compact.json # (architecture prepared, tokens ceholder)
| | ├── comfortable.json
| | └── spacious.json
| └── _future-modes/ # 🏗️ COMPLETE STRUCTURE PREPARED
| ├── responsive/ # (folder structure + placeholder f )
| ├── contrast/
| ├── motion/
| ├── typography/
| └── user-preferences/
` `
```

```
├── lightness/ # ✅ MVP: Existing (light/dark)
├── interaction-emphasis/ # ✅ MVP: Existing (s2-emph  s modes: low/standard/high)
├── viewport/ # ✅ MVP: Simplified (desktop/mobile on
├── component-size/ # ✅ MVP: Full implementation
| ├── sm.json
| ├── md.json
| └── lg.json
├── density/ # 🔄 STRUCTURE READY: Files created, imp  ntation deferred
└── _future-modes/ # 🏗 COMPLETE STRUCTURE: All folde   placeholder files
`
```

🎯 **Implementation Phases:**

- `Phase 1 ( VP): 4 essential modes -` lightness × int ion-emphasis × viewport × component-size
- `Phase 2: Add advanced system modes -` + density × ast × motion + full viewport coverage
- `Phase 3: Add user preference & accessibility moc` + typography × user-font-scale × user-motion-pref × user-contrast-pref × dyslexia-support

Complete architecture prepared from start, selective ivation by phase

✅ Already Implemented - Icon Holder Compone

Location: src/lib/themes/component/atom/icon_holder/ `

```
├── sm.json # Small variant
├── ium variant
├── ge variant
└── Context-independent
```

Curr :

```j
{
"ob
"c":
"ic
"siz
"st alue": "{ob.p.size.250}" },
"mir ": "{ob.p.size.175}" }
}
}
}
}
}
```

implem component

Modes

```
` json
{
"padding": {
"vertical": {
"sm": { "$value": "{ob.s.spacing.none}" },
"md": { "$value": "{ob.s.spacing.xs}" },
"lg": { "$value": "{ob.s.spacing.lg}" }
},
"horizontal": {
"sm": { "$value": "{ob.s.spacing.md}" },
"md": { "$value": "{ob.s.spacing.lg}" },
"lg": { "$value": "{ob.s.spacing.2xl}" }
}
}
}
`
```

Current Token Architecture

Existing Semantic Structure:

```
`
semantic/
```

```
├── sizing.json # ✅ W3C DTCG compliant dimension tok
├── spacing/
| ├── desktop.json # ✅ Responsive spacing with multi ers
| └── mobile.json # ✅ Responsive spacing with multip s
├── color/
| ├── s1-lightness/ # ✅ Theme mode implementation
| ├── s2-emphasis/ # ✅ Emphasis mode implementation
| └── s3-semantic/ # ✅ Compiled semantic layer
└── [other categories]
`
```

Global Theme System:

```
`
global/themes-user/
├── lightness/ # ✅ Light/dark theme switching
| ├── dark.json
| └── light.json
└── viewport/ # ✅ Responsive multiplier system
├── desktop.json # mult_responsive: 4
├── mobile.json # mult_responsive: 5
└── static.json
```

`

odes Architecture Implementatio Strategy

Phase 1: Establish Universal odes Archit ure Foundation

Create semantic/modes/ Architecture Framewor

`

```
semantic/
├── modes/ # 🆕 NEW: Universal modes system architect
| ├── component-size/ # 🆕 Component dimension modes
| | ├── sm.json # Small size mode tokens
| | ├── md.json # Medium size mode tokens (baseline)
| | └── lg.json # Large size mode tokens
| ├── density/ # 🆕 Spacing density modes
| | ├── compact.json # Tighter spacing multipliers
| | ├── comfortable.json # Standard spacing (baseline
| | └── spacious.json # Looser spacing multipliers
| ├── _future-modes/ # 🆕 Reserved space for modes re ing research
| | ├── responsive/ # Phase 2: Viewport-based scaling amework enabled, implementation deferred)
| | ├── contrast/ # Phase 2: A11y high contrast modes search required)
| | ├── typography/ # Phase ~~2: Font~~ scaling, reading s (research required)

| | └── user-preferences/ # Phase 3: User-controlled modes (font scaling, motion, contrast, dyslexia support)

| └── README.md # 🆕 Modes architecture system docume ion
```

Key odes Architecture Principles:

- **Scalable**: Easy to add new mode categories
- **Consistent**: All modes follow the same token struc patterns
- **Semantic**: Mode tokens reference primitive/semanti kens, not hardcoded values
- **Extensible**: Architecture supports future A11y, UX d brand mode requirements

Phase 2: Semantic ode Token Definitions

Component Size odes (semantic/modes/compone size/)

sm.json - Small component dimensions: ` json

```
{

"ob": {

"s": {

"modes": {

"component-size": {

"button": {

"height": { "$type": "dimension", "$value": "{ob.s.si lement.sm}" },

"min-height": { "$type": "dimension", "$value": "{ob. ze.element.sm}" }

},
"height": { "$type": "dimension", "$value": "{ob.s.si lement.sm}" }
},
"tag": {
"padding-vertical": { "$type": "spacing", "$value": " s.spacing.none}" },
"padding-horizontal": { "$type": "spacing", "$value": b.s.spacing.md}" }
}
}
}
}
}
}
```

` md.json - Medium component dimensions (baseline): json

```
{
"ob": {
"s": {
"modes": {
"component-size": {
"button": {
"height": { "$type": "dimension", "$value": "{ob.s.si lement.md}" },
"min-height": { "$type": "dimension", "$value": "{ob. ze.element.md}" }
"input": {
"height": { "$type": "dimension", "$value": "{ob.s.si lement.md}" }
},
"tag": {
"padding-vertical": { "$type": "spacing", "$value": " s.spacing.xs}" },
"padding-horizontal": { "$type": "spacing", "$value": b.s.spacing.lg}" }
}
}
}
}
}
}
```

`

Density odes (semantic/modes/density/)

`comfortable.json - Standard density (baseline):` ` js

```
{
"ob": {
"s": {
"modes": {
"density": {
```

```
"stack-gap": { "$type": "spacing", "$value": "{ob.s.s ng.lg}" },
"card-padding": { "$type": "spacing", "$value": "{ob. acing.xl}" },
"section-margin": { "$type": "spacing", "$value": "{o spacing.2xl}" }
}
}
}
}
}
}
```

` compact.json - Tighter density: ` json

```
{
"ob": {
"s": {
"modes": {
"density": {
"layout": {
"stack-gap": { "$type": "spacing", "$value": "{ob.s.s ng.md}" },
"card-padding": { "$type": "spacing", "$value": "{ob. acing.lg}" },
"section-margin": { "$type": "spacing", "$value": "{o spacing.xl}" }
}
}

}

}

}
```

`

Phase 3: Universal odes Architecture The Integration

Add to global/themes-user/ - Comprehensive s Support

`

```
global/themes-user/
├── lightness/ # ✅ Existing: Light/dark theme modes
├── viewport/ # ✅ Existing: Responsive scaling modes
├── component-size/ # 🆕 NEW: Size modes configuratio esign system controlled)
| ├── sm.json # References semantic/modes/component-s sm.json
| ├── md.json # References semantic/modes/component-s md.json (default)
| └── lg.json # References semantic/modes/component-s lg.json
├── density/ # 🆕 NEW: Density modes configuration (p ct designer controlled)
| ├── compact.json # References semantic/modes/densit mpact.json
| ├── comfortable.json # References semantic/modes/de y/comfortable.json (default)
| └── spacious.json # References semantic/modes/densi pacious.json
└── future-modes/ # 🆕 RESERVED: Modes architecture y for expansion (Phase 2+)
```

```
├── responsive/ # Phase 2: Viewport-based scaling mod (framework ready, research needed)
├── contrast/ # Phase 2: A11y high contrast modes (re ch required)
├── motion/ # Phase 2: Animation preference modes (en d/disabled/reduced)
├── typography/ # Phase 2: Typography scaling modes ( arch required)
└── user-preferences/ # Phase 3: User-controlled A11y es (font scaling, motion prefs, contrast, dyslexia support)
```

ulti-Dimensional odes Architecture:

- **Phase 1:** lightness × viewport × component-size × den (4D) - Full implementation
- **Phase 2:** + responsive × contrast × motion × typograph (8D) - Framework enabled, research required
- **Phase 3:** + user-preferences (9D+) - User-control A11y modes (font scaling, motion, contrast, dyslexia support)
- **Future:** Additional modes as requirements emerge

Phase 4: Component igration

Strategy A: Keep Current Icon Holder Pattern Recommended

Rationale: Already working, follows semantic boundari
Current icon_holder structure → Continue using separa iles

- sm.json md.json lg.json reference semantic es

- `Allows component-specific size boundaries`

Strategy B: igrate Tag to Semantic Referenc

Before (current):

` json

```
{

"padding": {

"vertical": {

"sm": { "$value": "{ob.s.spacing.none}" },

"md": { "$value": "{ob.s.spacing.xs}" },

"lg": { "$value": "{ob.s.spacing.lg}" }

}

}

}
```

`

After (semantic reference):

` json

```
{

"padding": {

"vertical": {
```

```
"md": { "$value": "{ob.s.modes.component-size.tag.padding-vertical}" },
"lg": { "$value": "{ob.s.modes.component-size.tag.padding-vertical}" }
}
}
}
```

`

Implementation Recommendations

✅ Preserve What Works

- **Keep icon_holder file-based structure** - Already follows semantic boundaries
- **Keep responsive multiplier system** - mult_responsive in viewport themes works well
- **Keep existing semantic/sizing.json** - W3C DTCG compliant and properly structured

🔄 igrate Gradually

Pre- igration: System Cleanup ⚠️

1. **Create dedicated migration branch** - Work on featur[illegible]e-modes-migration to isolate changes from current development 2. **Create new Figma file for migration branch** - Duplicate current design system file and associate with the migration branch

values 5. **Test current build process** - Confirm all ex ng components compile successfully

Priority 1: Create Universal odes Architect Foundation

1. Create `semantic/modes/` folder structure with exte ble modes architecture
2. Define component-size modes tokens (sm/md/lg)

3. Define density modes tokens (compact/comfortable/s ous)

4. Reserve `_future-modes/` structure for planned arch ture expansions

Priority 2: Component Integration with odes hitecture

1. Tag component → Reference semantic modes architect

2. Add component boundaries (sm/md/lg limits per comp t)

3. Validate grid alignment with multiplier system

Priority 3: Establish ulti-Dimensional ode rchitecture System

1. Add component-size and density modes configuration ntrolled by design system consumers/product designers)

2. Test multi-dimensional modes combinations (lightne viewport × component-size × density)

3. Document modes interaction patterns and architectu xtensibility

4. Validate scalability for future modes additions re ing research (responsive, contrast, motion, typography)

Note: Density modes are design system configurations by product designers, not user preferences.

⚠️ Key Considerations

Grid Alignment Validation

- Ensure mult_responsive (4px desktop, 5px mobile[illegible]eserves 4px grid
- Test all mode combinations maintain proper align[illegible]
- Validate that semantic mode tokens respect base [illegible]

Component Boundaries

Based on competitive analysis findings, most design s[illegible]ms (75%) use independent component sizing:

` json

```
// Components define their supported size ranges

{

"button": { "modes": ["sm", "md", "lg"] }, // Support[illegible]l modes

"navigation": { "modes": ["md"] }, // Fixed medium on[illegible]

"card": { "modes": ["md", "lg"] } // Medium and large[illegible]y

}
```

` **Note:** This pattern is rare in industry - only 2/8[illegible]lyzed systems use component size coordination.

odes Architecture Theme Resolution Order - [illegible]nsible Framework

Phase 1: lightness × viewport × component-size × density [illegible] - Full Implementation) Phase 2: + responsive × contrast × motion × typography (8D - Framework enabled, research require[illegible]ase 3: + user-preferences (9D+ - User-controlled A11y modes requiring extensive research)
Future: Additional modes as requirements emerge (nD)

Validation needed: Test that "last wins" theme resolu[illegible] scales with multi-dimensional modes architecture combinations.

Implementation Checklist

Phase 1: Universal odes Architecture Fou[illegible]tion

- [ ] Create semantic/modes/ folder structure wi[illegible]xtensible modes architecture design
- [ ] Define component-size modes tokens (sm/md/lg[illegible]
- [ ] Define density modes tokens (compact/comfort[illegible]/spacious)
- [ ] Reserve _future-modes/ architecture for Pha[illegible] modes (responsive, contrast, motion, typography)
- [ ] Document modes architecture system in README[illegible] with Phase 1/2 roadmap

Phase 2: ulti-Dimensional odes Theme In[illegible]ration

- [ ] Add global theme files for size and density m[illegible] (Phase 1 implementation)
- [ ] Update $themes.json to support multi-dimens[illegible]l modes architecture system
- [ ] Test modes configuration with existing light[illegible]/viewport modes
- [ ] Validate theme resolution scales with nD mod[illegible]ombinations
- [ ] Enable framework for Phase 2 modes (responsi[illegible] contrast, motion, typography) without full implementation

Phase 3: Component igration & odes Arch[illegible]ecture Validation

- [ ] Migrate tag component to semantic modes archi[illegible]ture references
- [ ] Validate icon_holder compatibility with new m[illegible] architecture
- [ ] Test grid alignment across all current modes [illegible]inations
- [ ] Document component integration patterns for f[illegible]e modes

Phase 4: odes Architecture Validation & [illegible]ure-Proofing

- [ ] Verify multiplier system maintains 4px grid a[illegible]ment
- [ ] Test all current modes combinations function [illegible]ectly
- [ ] Validate component boundaries prevent breakin[illegible]ints
- [ ] Document extensible patterns for future modes [illegible]hitecture categories
- [ ] **Phase 2 Planning**: Research responsive modes, [illegible] high contrast, animation preferences, and typography scaling
- [ ] **Phase 3 Planning**: Research user preference mo[illegible] (font scaling A+/A/A-, motion preferences, contrast settings, dyslexia-friendly typography)
- [ ] **Long-term Planning**: Define implementation roa[illegible] for user-controlled accessibility modes

Expected Benefits

For Designers

- Comprehensive mode system for component size coor[illegible]tion *(Note: Based on research, only 25% of design systems implement size coordination)*
- **Extensible architecture** supporting future mode ca[illegible]ries (contrast, motion, typography)
- Component size mode options (sm/md/lg) at the des[illegible]system level

For Developers

- **Universal modes architecture** that scales beyond s[illegible] and density
- Clear modes system patterns following W3C DTCG st[illegible]rds
- **Future-proof architecture foundation** for accessib[illegible]y and UX modes requirements

For System

- **Scalable tokens architecture** aligned with W3C DTC[illegible]andards
- Maintains existing responsive multiplier benefits
- Preserves 4px grid alignment system
- **Extensible modes architecture foundation** ready fo[illegible]anned mode categories (contrast, motion, typography)
- **ulti-dimensional modes theme system** supporting c[illegible]ex product requirements

🎯 VP Success Criteria

Architecture Validation

- [ ] Complete modes architecture structure prepare[illegible]r all future phases
- [ ] 4D mode system working: lightness × interaction [illegible]hasis × viewport × component-size
- [ ] Token resolution validates across all MVP mo[illegible]ombinations
- [ ] Theme switching works seamlessly between all [illegible] modes

Modes Architecture R[illegible]ap | Design System Team | September 2025

- `[ ] All 5 test components render correctly in all [illegible] mode combinations`

- `[ ]` infobox `shows all status colors work across [illegible]ghtness + interaction-emphasis modes`
- `[ ]` input text field `babuschka complexity handles [illegible]ted component modes correctly`
- `[ ] Size mode coordination works across` icon_ho[illegible] `,` button `,` tag` relationships

Technical Validation

- [ ] Simplified viewport (desktop/mobile) maintains 4px gri[illegible]nment
- [ ] Build system processes all mode combinations without [illegible]s
- [ ] No broken token references across any MVP mode com[illegible]tion
- [ ] Documentation clearly shows MVP scope vs future arch[illegible]ure vision

🔄 Post-MVP Expansion Path

Phase 2: Add density modes to existing 4D system → 5D syste[illegible]**ase 3**: Enable responsive, contrast, motion, typography → 9D system **Phase 4**: User preference modes → 12D system (complete)
Complete architecture prepared from MVP, selective activatio[illegible]phase