

Unitary Foundation

Delivery Scope & Commercial Terms

Version: 1.0

Last updated: [date]

Applies to: All Foundation engagements unless explicitly amended in writing

1. What This Document Is

This document defines:

What Unitary Foundation delivers

What inputs are required from the client

What is explicitly included and excluded

Commercial terms, limits, and acceptance criteria

This document is not:

A proposal

A customization offer

A consulting agreement

Reading this document does not initiate any work.

1. What Unitary Foundation Is

Unitary Foundation is a one-time technical delivery that converts an approved set of design values into a governed token system and platform-specific runtime output.

It establishes a deterministic design → code handoff for one platform and one theme.

1. Commercial Summary

Price: \$3,000 USD (fixed)

Delivery format: One ZIP archive

Timeline: 7–14 calendar days (after inputs are approved)

Platform: One (Web, iOS, or Android)

Theme: One (e.g. Light)

No subscriptions.

No retainers.

No hidden fees outside documented remediation paths.

1. Required Client Inputs

Work cannot begin until all required inputs are provided.

4.1 Design Source (Required)

One of the following:

A single approved Figma file

Equivalent design source clearly representing final intent

The source must represent current desired output, not exploratory work.

4.2 Token Volume Limit

Maximum 120 total token values, counted across all layers combined.

This includes:

Primitive tokens (raw values)

Semantic tokens (aliases)

Component appearance tokens

Shared structural tokens

If a semantic token references a primitive, both count separately.

Exceeding this limit requires an expanded scope (Tier 2).

4.3 Reference Components

Five (5) reference components must be agreed before work begins.

These are used only for verification.

Each component may include:

Up to 3 states (e.g. Default, Hover, Disabled)

One primary appearance

Variants, themes, or complex behaviors are out of scope.

1. Token Layers Defined

The Foundation scope supports the following layers only:

5.1 Primitive Tokens

Raw, unaliased values

Examples:

color.brand.500

space.8

font.size.16

5.2 Semantic Tokens

Aliases referencing primitives

Examples:

bg.page

text.primary

5.3 Component Appearance Tokens

Single appearance properties per component

Examples:

button.bg

card.text

5.4 Shared Structure Tokens

Geometry reused across components

Examples:

control.height

control.padding.x

1. What Is Delivered

6.1 Token System (Source of Truth)

tokens.json

Canonical, machine-readable token source

style-dictionary.config

Configuration for platform transforms

6.2 Platform Runtime Output (One Only)

Client must specify platform at order.

Web: tokens.css (CSS custom properties)

iOS: Tokens.swift (Swift constants)

Android:

Tokens.kt (Jetpack Compose), or

colors.xml (Android Views)

No cross-platform delivery in this tier.

6.3 Reference & Documentation

mapping-dictionary.md

Exact 1:1 mapping from design styles to tokens

integration-readme.md

Step-by-step instructions for developers

6.4 Verification Artifacts

For each reference component:

Figma instance using tokens only

Minimal code snippet demonstrating token usage

These are verification artifacts, not production components.

1. Quality Gate (Input Validation)

If required values are missing, ambiguous, or contradictory, a Quality Gate is triggered.

Work pauses until resolved.

Client may choose one option:

Provide missing values within 7 business days

Purchase Cleanup Pack

\$600 one-time

Up to 30 corrected token values

Approve hourly remediation

\$100 / hour

1. Stalled Project Policy

If inputs or approvals are not provided within 14 calendar days after a Quality Gate:

Project is placed on hold

Timeline pauses

Reactivation requires:

Submission of missing inputs or

\$250 reactivation fee (plus any remediation)

1. Acceptance Criteria

The engagement is considered complete when all of the following are true:

tokens.json generates runtime output without errors

All reference components render using tokens only

Code snippets compile without syntax errors

Mapping document is reviewed and approved

No unresolved Quality Gate conditions remain

1. Explicit Exclusions

The following are not included:

Design cleanup or UX changes

Dark mode or multi-theme support

Variant-heavy or state-heavy tokenization

Full component libraries

App or site integration

Build tool configuration

Cross-platform delivery

Any expansion requires Tier 2 or custom scope.

1. Intended Buyer Fit

Unitary Foundation is appropriate if:

You already have a defined visual direction

You want deterministic design → code output

You accept fixed limits and constraints

It is not appropriate if:

You want design help or iteration

You need multi-theme support

You expect bespoke flexibility

1. Next Step

If this scope fits your needs:

→ Request Audit

The audit determines whether your inputs qualify for this scope or require preparation.

End of Document