

Here is a comprehensive technical specification document designed for an AI developer (like Claude) to implement the **Hybrid ACT/CBT Architecture** into EchoVault.

This document prioritizes system stability, backwards compatibility with your existing main branch, and a clean user experience.

Technical Specification: Hybrid ACT/CBT Architecture Integration

1. Executive Summary

Objective: Evolve EchoVault from a strictly CBT-based (Cognitive Behavioral Therapy) journaling tool into a Hybrid system that intelligently integrates ACT (Acceptance and Commitment Therapy) and Mindfulness.

Core Philosophy: Mental health is dynamic. Users sometimes need to challenge thoughts (CBT) and sometimes need to accept feelings (ACT). The system will use an "AI Director" to select the best default framework while giving users the ultimate control to "Pivot" between them.

Scope of Changes:

1. **Backend (functions/index.js):** Update analyzeEntry prompt to support polymorphic output (CBT vs. ACT).
 2. **Database (Firestore):** Expand the analysis map in entries collection to support distinct act_analysis and cbt_breakdown objects.
 3. **Frontend (EntryCard.jsx):** Refactor card UI to handle dynamic frameworks and add a user toggle.
 4. **Frontend (ShelterView.jsx):** Integrate "Drop Anchor" mindfulness exercises.
-

2. Architecture & Data Schema

2.1 Database Schema (Firestore)

Collection: users/{userId}/entries

Field: analysis (Map)

We will introduce a **Non-Breaking Change** by making the schema additive. Existing fields (cbt_breakdown) remain to support legacy entries.

JavaScript

```
{  
  // EXISTING FIELDS (DO NOT REMOVE)  
  "mood_score": 0.45,  
  "framework": "cbt", // "cbt" | "act" | "celebration" | "general" (Added "act")  
  
  // EXISTING CBT STRUCTURE (Populated if framework == 'cbt')  
  "cbt_breakdown": {  
    "distortion": "Catastrophizing",  
    "challenge": "..."  
  },  
  
  // NEW ACT STRUCTURE (Populated if framework == 'act')  
  "act_analysis": {  
    "fusion_thought": "I am a failure", // The "hook"  
    "defusion_technique": "labeling", // "labeling", "visualization", "thanking_mind"  
    "defusion_phrase": "I'm having the thought that I am a failure",  
    "values_alignment": "Growth", // The value being challenged  
    "committed_action": "Read one page despite the fear."  
  }  
}
```

3. Backend Implementation Guide

File: functions/index.js

Target Function: analyzeEntry

3.1 Prompt Engineering Strategy

We will modify the system prompt to act as a "Clinical Director". It will analyze the entry and decide *which* tool to pull from the toolkit.

Logic Flow:

1. **Analyze Context:** Is the user stuck in a logic loop (CBT)? Or are they fighting a feeling/grieving (ACT)?
2. **Select Framework:** Choose cbt or act based on the analysis.
3. **Generate Output:** Return the specific JSON structure for that framework.

3.2 Code Modification (analyzeEntry function)

Update the prompt within analyzeEntry to include the ACT definition and routing logic.

New Prompt Logic:

JavaScript

```
const prompt = `Analyze this journal entry and route to the appropriate therapeutic framework.
```

ROUTING LOGIC:

1. "cbt" (Cognitive Behavioral): Use when user has specific "glitchy" logic, cognitive distortions (all-or-nothing), or requires fact-checking.
2. "act" (Acceptance & Commitment): Use when user is struggling with difficult *feelings* (grief, shame, anxiety) where "fighting" the feeling makes it worse. Focus on unhooking and values.
3. "celebration": If text describes wins/gratitude.
4. "general": Neutral updates.

Return JSON:

```
{  
  "title": "Short title",  
  "tags": ["tag1"],  
  "mood_score": 0.0-1.0,  
  "framework": "cbt" | "act" | "celebration" | "general",  
  
  // INCLUDE IF FRAMEWORK == 'cbt'  
  "cbt_breakdown": {  
    "automatic_thought": "The negative thought",  
    "distortion": "Label",  
    "perspective": "Reframe/Evidence"  
  },  
  
  // INCLUDE IF FRAMEWORK == 'act'  
  "act_analysis": {  
    "fusion_thought": "The thought the user is 'fused' with (taking as absolute truth)",  
    "defusion_phrase": "A phrase to create distance, e.g., 'I notice I'm having the thought that...'",  
    "values_context": "The core value at stake (e.g., Connection, Creativity)",  
    "committed_action": "A tiny, concrete step aligned with their values, not controlled by mood"  
  }  
}  
;
```

4. Frontend Implementation Guide

4.1 "Pivot" Capable Entry Card

File: src/components/entries/EntryCard.jsx

Requirement: The card must display the data provided by the backend but allow the user to see the "other side" if desired.

Note: For V1, the "Pivot" toggle will likely need to trigger a re-analysis if the other data isn't present, OR we purely rely on the AI's best judgment to keep costs low. We will implement the AI's judgment first.

Component Logic:

1. **Read Framework:** const mode = entry.analysis?.framework || 'general'.
2. **Render ACT Block:** Create a new visual style for ACT to distinguish it from CBT.
 - o *CBT Style:* Blue/Clinical/Logical (Tables, "Try thinking this").
 - o *ACT Style:* Teal/Organic/Flowing (Clouds, "Make space for").

New ACT UI Block Snippet:

JavaScript

```
{/* ACT Visualization */}
{mode === 'act' && actData && (
  <div className="bg-teal-50 rounded-xl p-4 border border-teal-100">
    <div className="flex items-center gap-2 mb-3">
      <Wind className="text-teal-600" size={16} />
      <span className="text-xs font-bold text-teal-700 uppercase">Defusion</span>
    </div>
    <div className="text-teal-900 text-sm mb-2">
      <span className="opacity-75">Instead of:</span>
      <span className="line-through decoration-teal-300">'{actData.fusion_thought}'</span>
    </div>
    <div className="text-teal-800 font-medium text-sm bg-white/50 p-2 rounded-lg">
      Try: "{actData.defusion_phrase}"
    </div>
    <div className="mt-3 pt-3 border-t border-teal-100 flex gap-2">
      <Compass size={14} className="text-amber-600" />
      <span className="text-xs text-amber-800">Value: {actData.values_context}</span>
    </div>
  </div>
)}
```

4.2 Interactive Shelter View

File: src/components/dashboard/views/ShelterView.jsx

Requirement: Add the "Grounding" (ACT) exercise alongside "Vent".

State Changes:

- Introduce internal state [mode, setMode] to switch between the default menu and the active exercise.

New "Drop Anchor" Component:

- **Step 1 (Acknowledge):** "Name the hardest feeling showing up right now." (Input/Mental check).
 - **Step 2 (Connect):** "Press your feet into the floor." (Animation/Timer).
 - **Step 3 (Engage):** "Name 3 things you see."
-

5. Deployment & Safety Strategy

5.1 Backwards Compatibility Check

- **Legacy Entries:** Old entries usually lack a framework field.
- **Fix:** In EntryCard.jsx, if entry.analysis.cbt_breakdown exists but framework is missing, default to cbt. If neither exists, default to general. This ensures the new code doesn't crash on old data.

5.2 Error Handling (Graceful Degradation)

- If the AI returns framework: 'act' but fails to generate act_analysis, the frontend should fallback to displaying the generic text and mood_score rather than crashing.
- **Implementation:** Use Optional Chaining (?) aggressively in the new React components.

5.3 Verification Steps (Post-Deploy)

1. **Create a "Logic" Entry:** Write "I'm useless because I didn't finish my todo list."
 - *Expectation:* System detects cbt and offers a Reframe.
 2. **Create a "Grief" Entry:** Write "I feel so heavy and sad about the news today."
 - *Expectation:* System detects act and offers a Defusion/Validation phrase.
 3. **View Old Entry:** Open an entry from last week.
 - *Expectation:* It still renders the old CBT data correctly.
-

6. Prompt for Claude

Copy and paste the following prompt to Claude to initiate the build:

"I need to implement a Hybrid ACT/CBT architecture for EchoVault. I have uploaded the current functions/index.js, src/services/analysis/index.js, EntryCard.jsx, and ShelterView.jsx.

Please strictly follow the **Technical Specification** below to implement these changes.

Critical Constraints:

1. **No Regressions:** Logic for existing CBT entries must remain exactly as is.
Use defaults to handle legacy data.
2. **Database:** Do not delete existing fields. Only add act_analysis and framework to the schema.
3. **Prompting:** Update the analyzeEntry system prompt in functions/index.js to intelligently route between 'cbt' and 'act' based on the routing logic provided in the spec.
4. **UI:** Update EntryCard.jsx to render the new ACT data structure when framework === 'act'.

Please generate the updated code for functions/index.js and EntryCard.jsx first."