Contents

Think Tank Platform Architecture

Advanced Multi-Step AI Problem Solving

Version 3.2.0 | December 2024

A comprehensive technical architecture document for the Think Tank AI reasoning platform

---

Table of Contents

---

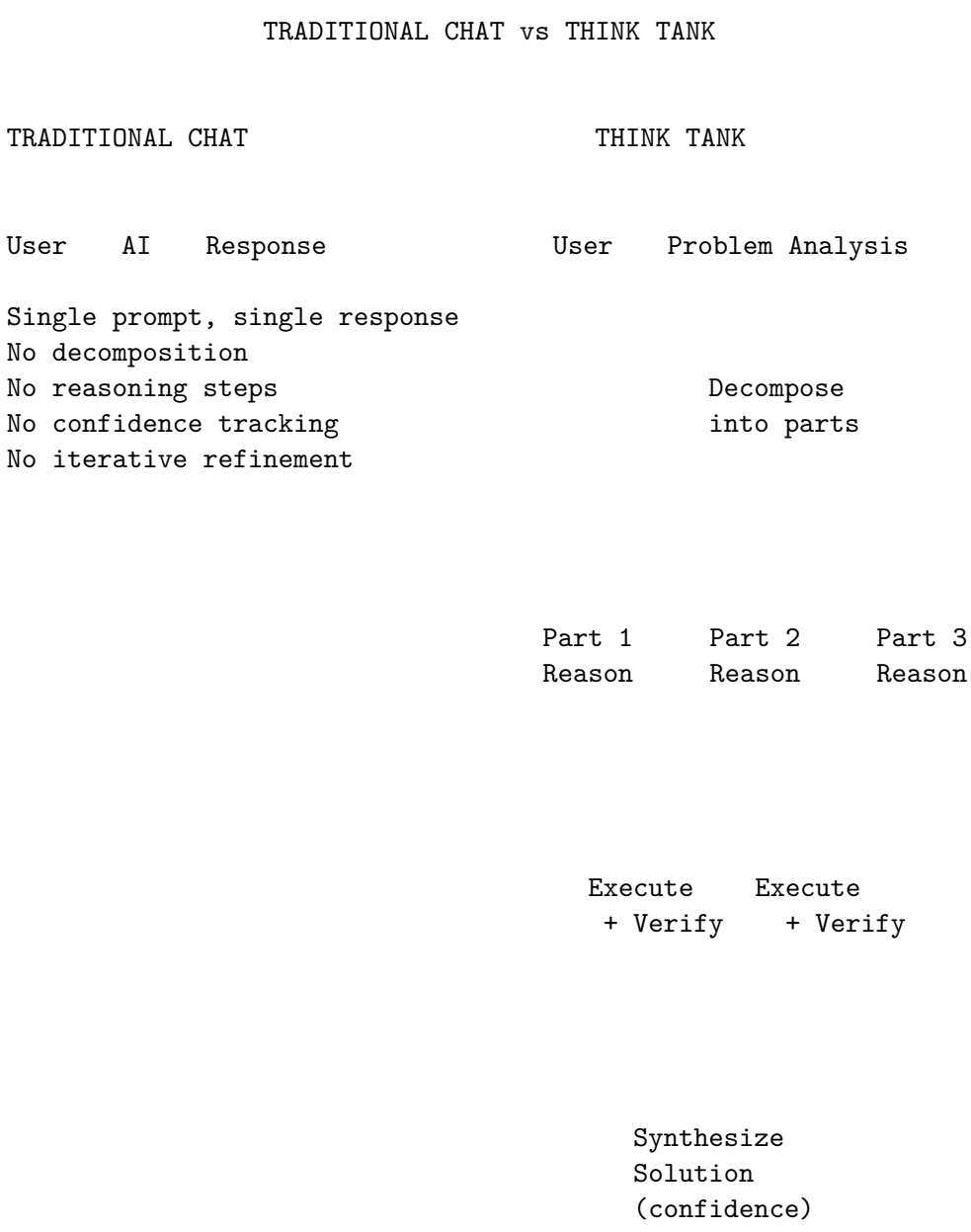
1. Platform Overview

1.1 What is Think Tank?

Think Tank is an advanced AI reasoning platform that decomposes complex problems into manageable sub-problems, applies multi-step reasoning, and synthesizes comprehensive solutions using orchestrated AI models.

Unlike simple chat interfaces, Think Tank: - **Decomposes** complex problems into sub-tasks - **Reasons** through each component step-by-step - **Executes** specialized AI calls for each step - **Synthesizes** results into coherent solutions - **Tracks** confidence and quality throughout

1.2 Think Tank vs Traditional Chat



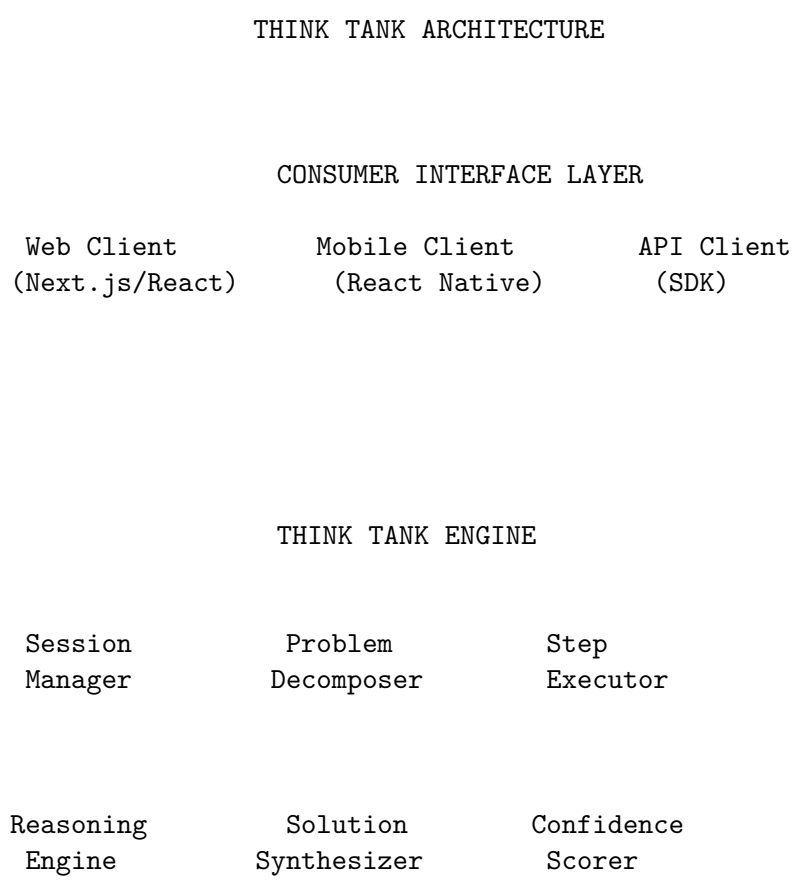
1.3 Key Capabilities

Capability	Description
Problem Decomposition	Breaks complex questions into manageable sub-problems
Multi-Step Reasoning	Chain-of-thought with recorded steps
Domain Specialization	8+ specialized reasoning modes
Confidence Tracking	Quality scores for every step
Artifact Generation	Code, documents, diagrams as outputs
Real-time Collaboration	Multiple users solving together
Session Persistence	Resume any session later
Cost Transparency	Token and cost tracking per step

---

2. Core Architecture

2.1 System Components



ORCHESTRATION LAYER

OrchestrationPatternsService

- 49 workflow patterns
- Parallel execution
- AGI model selection
- Mode-aware invocation

ModelRouterService

- 106+ AI models
- Live metadata
- Intelligent routing
- Fallback handling

DATA LAYER

Sessions	Conversations	Messages	Artifacts
(Aurora)	(Aurora)	(Aurora)	(S3)

2.2 Think Tank Engine

The core engine that powers intelligent problem solving:

THINK TANK ENGINE DETAIL

```
class ThinkTankEngine {

    async solve(problem: ThinkTankProblem): Promise<ThinkTankResult>

    1. CREATE SESSION
       • Initialize session with problem context
       • Set domain mode and preferences
       • Record start time and user info
```

2. DECOMPOSE PROBLEM
   - AI analyzes problem structure
   - Identifies sub-problems and dependencies
   - Creates execution plan

3. FOR EACH SUB-PROBLEM:

   a. REASON
      - Chain-of-thought analysis
      - Record reasoning steps

   b. EXECUTE
      - Call appropriate AI model(s)
      - May use parallel execution
      - Track tokens and cost

   c. RECORD STEP
      - Save step result with confidence
      - Update session state

4. SYNTHESIZE SOLUTION
   - Combine all step results
   - Generate final answer with reasoning
   - Calculate overall confidence

5. RETURN RESULT
   - Solution with confidence score
   - All recorded steps
   - Total cost and token usage

}

---

3. Problem Solving Pipeline

3.1 Pipeline Stages

PROBLEM SOLVING PIPELINE

USER INPUT

"Design a scalable microservices architecture for an e-commerce

platform that handles 10M daily users with real-time inventory"

STAGE 1: PROBLEM ANALYSIS

- Identify problem type: System Design
- Detect domain: Engineering/Architecture
- Assess complexity: High
- Select domain mode: Engineering Mode
- Choose orchestration pattern: Decomposed Prompting

STAGE 2: DECOMPOSITION

Sub-Problem 1: Requirements Analysis
Sub-Problem 2: Service Identification
Sub-Problem 3: Data Architecture
Sub-Problem 4: Communication Patterns
Sub-Problem 5: Scalability Design
Sub-Problem 6: Infrastructure

Dependencies: [1] → [2,3] → [4] → [5] → [6]

STAGE 3: STEP-BY-STEP EXECUTION

Step 1: Requirements
  Model: Claude 3.5 (thinking mode)
  Tokens: 2,450 Cost: $0.024 Confidence: 0.92
  Output: Detailed requirements document

Step 2: Service Identification
  Model: GPT-4o + Claude (parallel, merge synthesis)
  Tokens: 3,200 Cost: $0.041 Confidence: 0.89
  Output: 12 microservices identified with boundaries

[Steps 3-6 continue...]

STAGE 4: SYNTHESIS

- Combine all step outputs
- Generate comprehensive solution document
- Include architecture diagram (artifact)
- Validate consistency across steps
- Calculate final confidence: 0.88

FINAL OUTPUT

- Complete microservices architecture document
- Service interaction diagrams
- Database schema recommendations
- Infrastructure as code templates
- Scaling strategies and benchmarks

Total: 12,400 tokens $0.18 6 steps 45 seconds

3.2 Step Recording

Every reasoning step is recorded with comprehensive metadata:

STEP RECORD STRUCTURE

```
interface ThinkTankStep {
  stepId: string;                    // Unique step identifier
  sessionId: string;                 // Parent session
  stepOrder: number;                 // Execution order
  stepType: StepType;                // decompose | reason | execute | ..
  title: string;                     // Human-readable step name
  description: string;               // What this step does

  // Execution Details
  input: {
    prompt: string;                  // Input to AI
    context: Record<string, any>;    // Previous step outputs
    parameters: Record<string, any>; // Step-specific params
  };
```

```
  output: {
    response: string;                // AI response
    artifacts: Artifact[];           // Generated files/diagrams
    structuredData?: any;            // Parsed structured output
  };

  // Model & Cost
  modelUsed: string;                 // Which AI model
  modelMode: ModelMode;              // thinking | fast | creative | ..
  tokensUsed: number;                // Total tokens
  costCents: number;                 // Cost in cents
  latencyMs: number;                 // Execution time

  // Quality
  confidence: number;                // 0-1 confidence score
  reasoning: string;                 // Explanation of confidence

  // Parallel Execution (if applicable)
  wasParallel: boolean;
  parallelModels?: string[];         // Models used in parallel
  synthesisStrategy?: string;        // How results were combined

  // Timestamps
  startedAt: Date;
  completedAt: Date;
}
```

---

4. Session Management

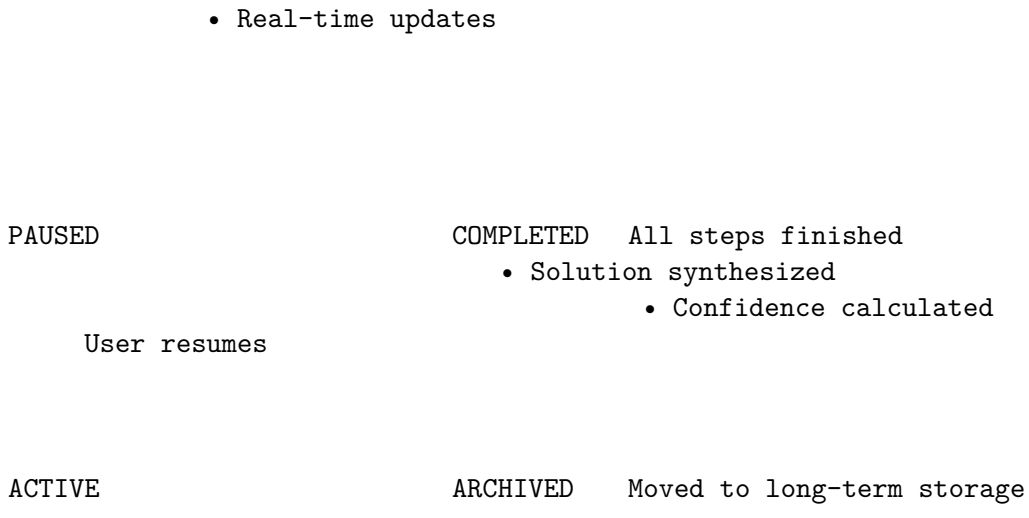
4.1 Session Lifecycle

SESSION LIFECYCLE

NEW User starts a new problem-solving session

ACTIVE Session is being worked on

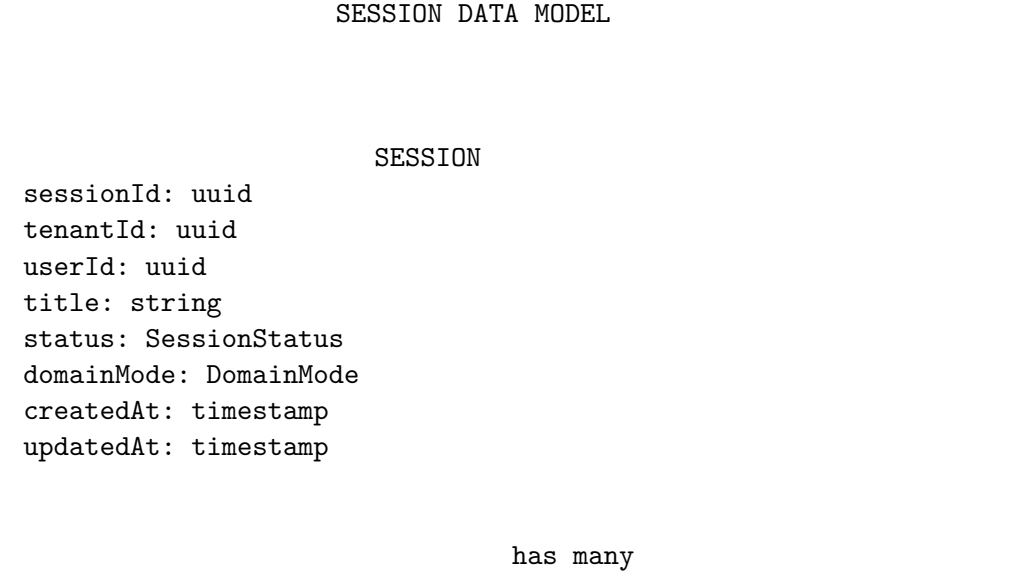
- Steps executing
- User can interact



Session States:

- NEW - Just created, no work done
- ACTIVE - Currently processing or awaiting input
- PAUSED - User paused, can resume
- COMPLETED - All steps done, solution ready
- ARCHIVED - Moved to cold storage
- FAILED - Unrecoverable error occurred

4.2 Session Data Model



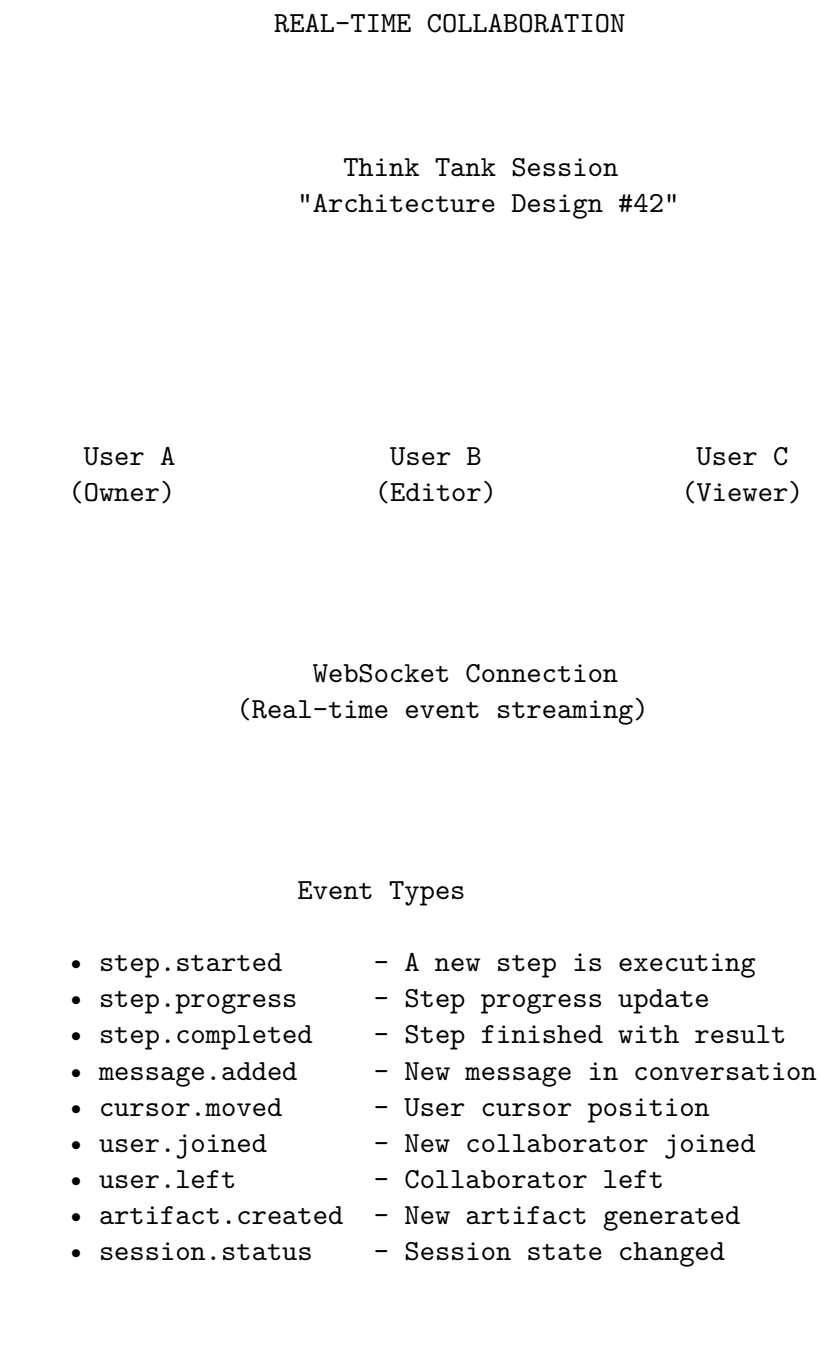
```
                    CONVERSATIONS
conversationId: uuid
sessionId: uuid (FK)
title: string
createdAt: timestamp

                              has many

                     MESSAGES
messageId: uuid
conversationId: uuid (FK)
role: 'user' | 'assistant' | 'system'
content: text
createdAt: timestamp

                              has many

                      STEPS
stepId: uuid
sessionId: uuid (FK)
stepOrder: integer
stepType: StepType
input: jsonb
output: jsonb
modelUsed: string
tokensUsed: integer
costCents: decimal
confidence: decimal
startedAt: timestamp
completedAt: timestamp

                              has many

                     ARTIFACTS
artifactId: uuid
stepId: uuid (FK)
type: 'code' | 'diagram' | 'document' | 'data'
filename: string
mimeType: string
s3Key: string
sizeBytes: integer
```

```
createdAt: timestamp
```

---

5. Collaboration Features

5.1 Real-Time Collaboration



COLLABORATION ROLES:

Role	Permissions
Owner	Full control, manage collaborators, delete session
Editor	Add messages, trigger steps, view all content
Viewer	Read-only access to session and results
Commenter	View + add comments, no step triggering

---

6. Domain Modes

6.1 Specialized Reasoning Modes

DOMAIN MODES

Think Tank adapts its reasoning approach based on problem domain:

RESEARCH MODE

Best for: Academic research, literature review, fact-finding
Models: Perplexity Sonar, Claude (deep_research mode)
Features:
- Source citation
- Cross-reference verification
- Comprehensive literature synthesis

ENGINEERING MODE

Best for: System design, architecture, technical problems
Models: Claude, GPT-4o, DeepSeek (code mode)
Features:
- Code generation as artifacts
- Architecture diagrams
- Technical trade-off analysis

ANALYTICAL MODE

Best for: Data analysis, math, statistics, quantitative problems
Models: o1, Claude (thinking mode), DeepSeek R1
Features:
• Step-by-step mathematical reasoning
• Statistical analysis
• Proof verification

CREATIVE MODE

Best for: Writing, brainstorming, ideation, design
Models: Claude, GPT-4o (creative mode, high temperature)
Features:
• Multiple creative alternatives
• Iterative refinement
• Style adaptation

LEGAL MODE

Best for: Contract analysis, compliance, legal research
Models: Claude (precise mode), GPT-4o
Features:
• Citation of legal precedents
• Risk assessment
• Compliance checking

MEDICAL MODE (HIPAA Compliant)

Best for: Clinical analysis, medical research (non-diagnostic)
Models: Claude (precise mode), approved medical models
Features:
• PHI sanitization
• Medical literature citation
• Disclaimer generation

BUSINESS MODE

Best for: Strategy, planning, market analysis, business problems

Models: GPT-4o, Claude, Gemini
Features:
• Framework application (SWOT, Porter's, etc.)
• Financial modeling
• Competitive analysis

GENERAL MODE

Best for: Mixed problems, general questions
Models: Automatically selected based on sub-problem analysis
Features:
• Dynamic mode switching per step
• Balanced approach

---

7. Quality & Confidence

7.1 Confidence Scoring System

CONFIDENCE SCORING SYSTEM

Every step and the final solution receives a confidence score (0-1):

CONFIDENCE FACTORS

Factor	Contribution
Model Agreement	+0.2 if parallel models agree
Reasoning Depth	+0.15 for thorough chain-of-thought
Source Quality	+0.15 for cited/verified sources
Task Complexity	-0.1 for very complex sub-problems
Model Confidence	+0.1 for high model self-confidence
Consistency	+0.1 for consistency with prior steps
Verification	+0.2 if verified by second model

CONFIDENCE LEVELS

```
0.9 - 1.0                VERY HIGH  - Strong consensus
0.7 - 0.9                  HIGH       - Reliable
0.5 - 0.7                    MODERATE   - Review recommended
0.3 - 0.5                      LOW        - Uncertain
0.0 - 0.3                        VERY LOW   - Needs verification
```

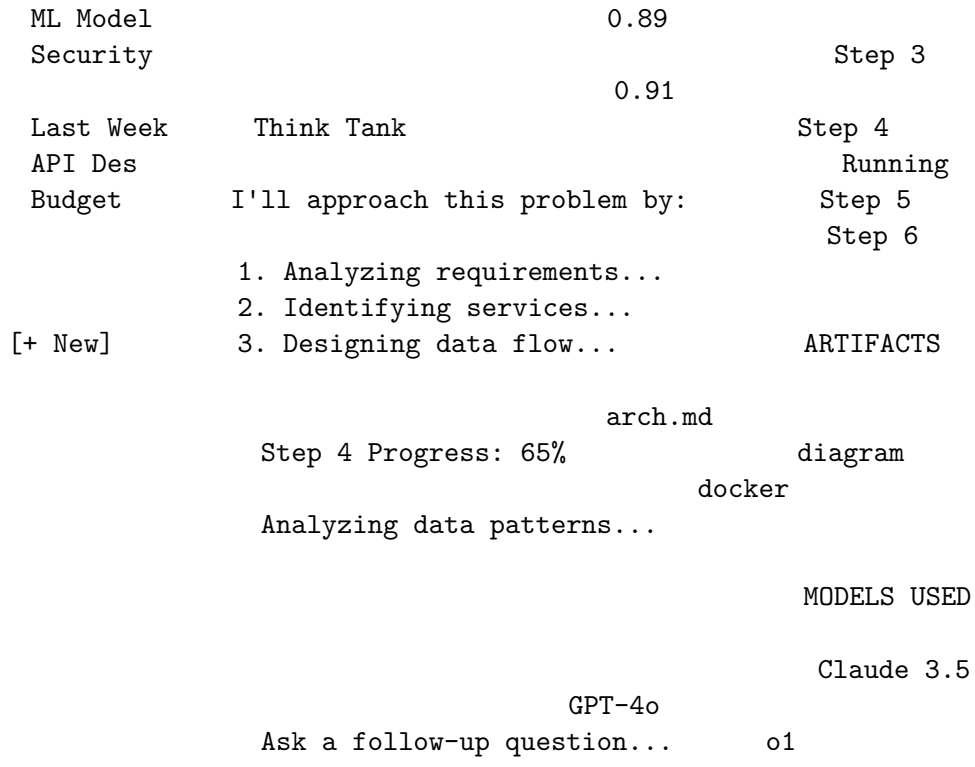
```
FINAL SOLUTION CONFIDENCE

Formula:

final_confidence = weighted_avg(step_confidences) × synthesis_factor

Where:
• step weights based on importance/complexity
• synthesis_factor accounts for integration quality
```

---

8. User Interface

8.1 Think Tank UI Layout

```
                    THINK TANK USER INTERFACE


      Think Tank                          [New] [Share] [Export]
   Logo     Problem: "Design microservices architecture..."
      Mode: Engineering   Confidence: 0.88   Cost: $0.18


SESSIONS              MAIN CONVERSATION                DETAILS

 Today                                 STEPS
 Arch #42       You
 Data Q        Design a scalable microservices       Step 1
                architecture for an e-commerce         0.92
 Yesterday     platform that handles 10M...          Step 2
```



Think Tank Platform Architecture v3.2.0

Advanced AI reasoning for complex problems
