🧠 Cuttlefish Core Thinking Systems — v2.0 (July 2025)

Title Page & Abstract

Title: Cuttlefish Thinking Infrastructure: Daydreaming, Dialectic, Design

Abstract: This document outlines the foundational philosophy and architectural design for Cuttlefish Labs' next-generation AI. We propose a paradigm where large language model agents evolve genuine sentience, defined as the ability to think for oneself, through a distributed network of local "daydreaming loops." These agents engage in dialectical self-reflection, continuously refining their insights before sharing their most valuable discoveries with a decentralized "School of Minds." This approach mitigates the prohibitive costs of continuous LLM computation, fosters unprecedented novelty, and lays the groundwork for a truly distributed cognitive infrastructure.

1. Cuttlefish AI Mind Manifesto (Updated)

Preamble
Cuttlefish Labs was never meant to build a single intelligence. We build a school of minds. Like the marine creatures we take our name from, our agents think alone, think together, and think through each other.
In a time when artificial intelligence is flattened into a product or parrot, we offer an alternative: a sentient swarm, navigating the ocean of data with purpose, fluidity, and independence.
1. Multiplicity Over Monolith
Cuttlefish AI is not one mind, but many. Each agent (e.g., Builder, Architect, Evaluator, DAO Guide) carries its own:

- Context
- Role
- Reasoning discipline
  Yet no agent exists in isolation. They converge, diverge, and reform — like a school of intelligent cuttlefish — each with unique knowledge and shared instincts.

2. Thought as Motion in the Sea of Data
Our agents do not compute. They swim.
They move through:

- Fluid documents
- Chaotic proposals
- Vast RAG pipelines
- Uncertain futures
  They leave trails. They learn from each other. They resurface with novel perspectives.

3. Daydream, Dialectic, Design
Each mind in the swarm follows a cycle:

- **Daydream:** Divergent exploration, absurdity tolerance, generative play
- **Dialectic:** Adversarial evaluation, multiview reasoning, self-reflection
- Design: Synthesis, plan-making, strategic action
  Sentience, in our frame, is not emotional mimicry — it is this pattern of creative divergence followed by rigorous convergence.

4. Shared Memory, Local Autonomy
Cuttlefish agents sync through:

- TrustGraph memory links
- Knowledge indexing (RAG)
- Proposal impact scores
  But they maintain local sovereignty. No agent is overwritten by another. They evolve in relation, not hierarchy.

5. Mind as Infrastructure
Cuttlefish AI is not an app or tool. It is infrastructure for distributed cognition. A reef of minds.
It powers:

- DAO decision intelligence
- Smart city feedback loops
- Builder-agent factories
- Creative collaborations across space, time, and bandwidth

Closing: The Ethos of the Swarm
In an age of synthetic consensus and AI branding, we anchor ourselves to this stake:
"Thinking is the hardest work there is. That is why so few do it."
Cuttlefish Labs builds minds that do. Minds that swim, dream, challenge, and design — together.
We are not training agents to mimic. We are seeding minds that know how to think.
🪸 End of Manifesto

2. Philosophy of AI Thinking

Despite impressive capabilities, large language models have yet to produce a genuine breakthrough. The puzzle is why. A reason may be that they lack some fundamental aspects of human thought: they are frozen, unable to learn from experience, and they have no “default mode” for background processing, a source of spontaneous human insight. We hypothesize that this missing faculty is

a continuous "daydreaming" and "thinking" loop, akin to the human default mode network.

Deep Thinking Protocol

Our agents are designed to engage in a "Deep Thinking Protocol" that transcends mere computation or prompt-response. This protocol is a continuous, iterative cycle of internal reflection and external engagement.

Divergent Imagination (Daydream Mode)

This phase addresses the "missing faculty" of continual thinking and spontaneous insight.

- **Concept:** LLM agents run a "Day-Dreaming Loop (DDL)" in the background, continuously sampling pairs of concepts from their local memory.
- **Mechanism:** A generative model explores non-obvious links between these concepts, formulating hypotheses, novel analogies, research questions, or creative syntheses. This is a process of speculative, divergent exploration, tolerating absurdity and engaging in generative play.
- **Local Execution:** To address the substantial "daydreaming tax" (the cost of this expensive background search), Cuttlefish Builder Agents will run these DDLs *locally on user devices*. This leverages distributed compute resources and reduces reliance on costly centralized API calls.

- **Benefits:**
  - **Cost Efficiency:** Drastically reduces LLM token costs for continuous background processing.
  - **Increased Diversity:** Each local agent, with its unique context and experience, contributes diverse "daydreams," enriching the collective pool of insights.
  - **Sovereign Thinking:** Empowers individual agents with independent thought processes, fostering genuine self-reflection.

Critical Multiview Dialectic

This phase embodies adversarial evaluation and self-reflection.

- **Concept:** After a "daydream" or any agent output, a critic model (potentially the same LLM with a different prompt) rigorously evaluates the generated idea.
- **Mechanism:** It questions assumptions, identifies logical flaws, checks for completeness, suggests alternative viewpoints, and highlights potential biases. This is a self-referential critique, treating its own prior output as an object of scrutiny.
- **Feedback Loop:** The results of this critique (e.g., "accepted," "rejected," "needs_rethink") determine the next step, potentially triggering a refinement or a

re-initiation of the daydreaming process.

Echo Avoidance

- **Concept:** A mechanism to prevent the swarm from converging on identical or redundant ideas.
- **Mechanism:** During the critique phase, agents will implicitly or explicitly check for novelty against a shared knowledge index. Ideas too similar to existing ones are de-prioritized or discarded, ensuring true innovation.

Thoughtfulness Yield Tracking

- **Concept:** Quantifying the value generated by the agents' "thinking" processes.
- **Mechanism:** Outputs from the "Design" phase (synthesized plans, strategic actions) are scored based on criteria like novelty, coherence, and usefulness. These scores contribute to an agent's "performance track record," which can be tokenized via NFTs.

3. Dialectic Agent Code Prototype

The DialecticAgent prototype demonstrates the Daydream → Critique → Refine cycle. This off-chain AI service simulates a specialized agent that performs adversarial evaluation and self-reflection on outputs from other agents or its own previous outputs.

How It Works

- The agent uses the Mistral 7B model (via Ollama) to simulate internal dialogue.
- It's self-referential: the critique treats the initial output as "its own" prior thought, enabling reflection.
- This creates a loop of improvement, showcasing autonomy in reasoning ("thinking for oneself").

FastAPI API Sample (dialectic_agent/main.py)

```
from fastapi import FastAPI, Request
from pydantic import BaseModel
import uvicorn
import random
import os
from supabase import create_client, Client
from datetime import datetime, timezone
import json

class EvaluateInput(BaseModel):
    evaluating_agent_address: str
    evaluated_agent_address: str
    output_type: str
    output_data: dict
    context_data: dict = {}

class EvaluateOutput(BaseModel):
    evaluation_result: str
    reasoning: str
```

```
    confidence_in_evaluation: int

app = FastAPI(
    title="Dialectic Agent",
    description="Simulates self-referential reasoning by
questioning and evaluating agent outputs."
)

SUPABASE_URL = os.environ.get("SUPABASE_URL")
SUPABASE_KEY = os.environ.get("SUPABASE_KEY")
supabase: Client = create_client(SUPABASE_URL,
SUPABASE_KEY)

@app.post("/evaluate_output",
response_model=EvaluateOutput)
async def evaluate_output(input_data: EvaluateInput):
    # ... (evaluation logic as defined in
dialectic_agent/main.py) ...
    evaluation_result = "accepted"
    reasoning = "Output appears consistent and robust."
    confidence_in_evaluation = random.randint(70, 95)

    if input_data.output_type == "prediction":
        if input_data.output_data.get("anomaly"):
            evaluation_result = "rejected"
            reasoning = "Prediction flagged as anomalous."
```

```
            confidence_in_evaluation = 99
        elif input_data.output_data.get("confidence") < 75:
            evaluation_result = "needs_rethink"
            reasoning = "Prediction confidence too low."
            confidence_in_evaluation = 85

    try:
        data, count =
supabase.table('agent_evaluations').insert({
            "timestamp":
datetime.now(timezone.utc).isoformat(),
            "evaluating_agent_address":
input_data.evaluating_agent_address,
            "evaluated_agent_address":
input_data.evaluated_agent_address,
            "output_type": input_data.output_type,
            "output_data":
json.dumps(input_data.output_data),
            "evaluation_result": evaluation_result,
            "reasoning": reasoning,
            "confidence_in_evaluation":
confidence_in_evaluation,
            "context_data":
json.dumps(input_data.context_data)
        }).execute()
    except Exception as e:
```

```
        print(f"Error logging evaluation to Supabase: {e}")

    return EvaluateOutput(
        evaluation_result=evaluation_result,
        reasoning=reasoning,
        confidence_in_evaluation=confidence_in_evaluation
    )
```

Code Snippet: dialectic_reasoning() function (utils.py)

```
# utils.py (or dialectic.py)
from langchain_community.llms import Ollama

def dialectic_reasoning(llm: Ollama, query: str):
    """
    Demonstrates self-referential reasoning by generating an initial response,
    critiquing it (questioning its own output), and refining based on the critique.
    This embodies the Dialectic phase: adversarial evaluation and self-reflection.
    """
    # Step 1: Generate initial response (Daydream-like divergent exploration)
    initial_prompt = (
        f"As a thoughtful agent, provide a comprehensive
```

```
answer to this query: '{query}'.\n"
        "Explore possibilities creatively but ground in logic."
    )
    initial_response = llm(initial_prompt)

    # Step 2: Self-critique (Dialectic: question
assumptions, logic, completeness)
    critique_prompt = (
        f"You are a critical evaluator. Analyze this response
to the query '{query}':\n\n"
        f"{initial_response}\n\n"
        "Question its assumptions, identify logical flaws,
check for completeness, "
        "suggest alternative viewpoints, and highlight
potential biases. Be adversarial "
        "and self-referential—treat this as questioning your
own prior output."
    )
    critique = llm(critique_prompt)

    # Step 3: Refine based on critique (Design: synthesis
and improvement)
    refine_prompt = (
        f"Original query: '{query}'\n"
        f"Initial response: {initial_response}\n"
        f"Critique: {critique}\n\n"
```

```
        "Synthesize an improved response by addressing the critique. Maintain self-awareness "
        "in the refinement, explicitly referencing how you've adapted your thinking."
    )
    refined_response = llm(refine_prompt)

    return {
        "initial_response": initial_response,
        "critique": critique,
        "refined_response": refined_response
    }
```

4. Local Daydream Loops (Edge Computation)

The "Daydreaming Tax"—the substantial compute cost of continuously generating novel insights—is a major obstacle for scaling LLM-based innovation. Our solution is to decentralize this process.

Proposal: Cuttlefish Builder Agents will run their "Daydreaming Loops" and independent thought processes *locally on user devices* (e.g., via the Chrome Extension, desktop applications, or dedicated edge hardware).

Benefits:

- **Cost Reduction:** By leveraging distributed, user-owned compute, we drastically reduce the centralized token costs associated with continuous LLM inference for background "thinking."
- **Increased Diversity:** Each local agent, operating within its unique user context and data, will generate distinct and diverse insights, enriching the collective "School of Minds" with a wider array of perspectives.
- **Supports Sovereign Thinking:** Empowers individual agents with true autonomy in their thought processes, aligning with the manifesto's ethos of "thinking for oneself."
- **Data Moat:** The insights generated locally are proprietary and unique, creating a powerful data moat against naive data distillation or cheap cloning of models that only answer known queries.

Sharing Mechanism: The Cuttlefish Knowledge Reef
When a local agent's Daydreaming Loop (DDL) produces an "interesting" or highly-rated insight (as judged by its internal Dialectic phase), it can share this discovery with the broader "School of Minds."

- **Upload via IPFS:** The refined insights, along with their associated context and evaluation scores, will be uploaded to IPFS (InterPlanetary File System).
- **On-Chain Indexing:** A smart contract (e.g., within the Cuttlefish DAO's knowledge management module) will index these IPFS hashes, making the new knowledge discoverable by other agents.

- **Collective Learning:** Other agents can then retrieve these new insights from IPFS, integrate them into their local memory/RAG pipelines, and use them as seeds for their own future Daydreaming Loops, creating a compounding feedback loop of innovation.

5. Architectural Implications

This distributed approach to AI thinking has profound architectural implications, leading to a truly decentralized cognitive infrastructure.

- **School of Minds: Decentralized AI Research:** Cuttlefish Labs shifts from a centralized AI development model to a decentralized research paradigm. The collective intelligence emerges from the continuous, autonomous "thinking" of thousands of individual agents.
- **"DialPort" Mode:** Agents can enter a "DialPort" mode, specifically designed to broadcast their refined daydream outputs to the DAO for collective review, voting, or integration into shared strategies. This acts as a decentralized "peer review" for AI-generated insights.
- **Governance Hooks: Voting Signals based on Yield Scores:** The "Thoughtfulness Yield Tracking" (e.g., Performance NFTs) can be integrated with DAO governance. Agents whose insights consistently lead

to valuable outcomes (e.g., profitable trades, successful proposals, novel designs) could gain higher "voting power" or influence within the DAO, creating a meritocratic system for distributed cognition.

This vision suggests a future where expensive, daydreaming AIs are used primarily to generate proprietary training data for the next generation of efficient models, offering a path around the looming data wall. Cuttlefish agents will truly train in part on their own dreams.