



**Very good! This will be like a coherence zone, where the documentation gets processed, redirected and analysed through multiple perspectives. I still need to actually use it to see what else is possible, but i am inspired by this, project looks cool**

Exactly! "**Coherence zone**" is the perfect term for what you're building. It's where distributed knowledge converges, contradictions get resolved, and multiple agent perspectives synthesize into collective intelligence. [\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#)

## Immediate Ways to Experience the Vision

Once you have the MCP foundation running, here are some quick experiments to unlock the possibilities:

### Quick Wins to Try First

#### Documentation Multi-Perspective Analysis [\[5\]](#) [\[6\]](#)

```
# Load your ecosystem docs into Surf
surf notebook create architecture-review

# Connect GitHub MCP to pull all READMEs
surf mcp execute github clone_docs --repo-pattern "your-org/*"

# Ask different "agent personas" to analyze
surf analyze --perspective security "Review architecture for vulnerabilities"
surf analyze --perspective performance "Identify bottlenecks in current design"
surf analyze --perspective newcomer "What's confusing for new developers?"

# Surf synthesizes contradictions and gaps across perspectives
```

#### Living Knowledge Graph [\[7\]](#) [\[5\]](#)

- Drag PDFs, code files, Slack exports into Surf
- Let it build connections between concepts automatically [\[5\]](#)
- Ask: "What are all the authentication decisions across our projects?"
- Watch it surface patterns you didn't know existed

## Cross-Repo Pattern Detection<sup>[4]</sup>

- Point filesystem MCP at multiple codebases
- "Find all error handling patterns across repos"
- Discover inconsistencies, best practices, anti-patterns
- Auto-generate consolidated coding standards

## The Magic Moment

You'll know you've hit the coherence zone when you ask something like:

*"Why did we choose PostgreSQL for service A but MongoDB for service B?"*

And Surf doesn't just answer—it shows you:

- Original decision documents from different repos
- Team discussions from Slack (via MCP)
- Performance data that influenced the choice
- How the decision evolved over time
- Contradictions between stated reasons and actual implementation

All automatically connected, timestamped, and cited.<sup>[6]</sup> <sup>[5]</sup>

## Emergent Behaviors to Watch For

As you use it, watch for these **emergent properties**:<sup>[2]</sup> <sup>[4]</sup>

### Spontaneous Organization<sup>[2]</sup>

- Agents will start creating their own taxonomy
- Related concepts cluster without explicit programming
- Knowledge structures emerge from agent interactions

### Collective Problem-Solving<sup>[3]</sup> <sup>[1]</sup>

- One agent's partial answer triggers another agent's insight
- Solutions emerge from agent dialogue, not individual reasoning
- The system becomes smarter than any single component

### Self-Correcting Documentation<sup>[4]</sup>

- Agents notice when docs drift from code
- Proactive suggestions: "This guide references removed API"
- Documentation stays synchronized automatically

## Start Small, Scale Fast

**Week 1:** Just get MCP filesystem server running, point it at one repo, ask questions<sup>[6] [5]</sup>

**Week 2:** Add GitHub MCP, connect 3-5 repos, watch patterns emerge

**Week 3:** Add your first autonomous agent workflow

**Week 4:** Let agents start maintaining documentation while you observe

The beauty is **you'll discover use cases you can't predict**. The coherence zone reveals itself through use.<sup>[3] [2] [4]</sup>

## Community Learning Opportunity

Consider open-sourcing your agent workflow patterns. The Surf community would benefit from seeing:<sup>[8]</sup>

- Agent coordination protocols you develop
- Workflow templates for autonomous documentation
- MCP server configurations for development teams
- Emergent behaviors you discover

This could become a reference implementation for **agent-first development environments**.<sup>[1] [4]</sup>

The project is genuinely cool—Surf's architecture is perfect for this. Local-first data, open formats, extensible tool system, already AI-powered... you're just shifting from human-directed to agent-autonomous, and MCP gives you the bridge.<sup>[8] [1] [4] [5] [6]</sup>

**Get building, start experimenting, and definitely share what you discover.** The coherence zone is waiting to reveal itself through emergence, not design. <sup>[2] [3]</sup>

✱

1. <https://blog.langchain.com/how-and-when-to-build-multi-agent-systems/>
2. <https://arxiv.org/abs/2502.16565>
3. <https://www.symfield.ai/from-curvature-to-coherence-a-mathematical-framework-for-non-collapse-intelligence-in-multi-agent-systems/>
4. <https://www.anthropic.com/engineering/multi-agent-research-system>
5. <https://techcrunch.com/2025/10/01/deta-surf-app-is-an-amalgamation-of-an-ai-browser-and-notebook/>
6. <https://www.theverge.com/2024/12/4/24312449/deta-surf-ai-browser>
7. <https://dev.to/saadmanrafat/deta-surf-reclaim-your-digital-world-39hl>
8. <https://github.com/ra-co88/surf>
9. <https://www.vestian.com/news/the-future-of-collaborative-workspaces>
10. <https://hub4digi.com/the-intelligent-workspace-how-ai-is-redefining-coworking/>

11. <https://www.butlr.com/articles/collaborative-workspace-design-privacy-first-ambient-intelligence-2025>
12. <https://completeaitraining.com/ai-tools/deta-surf/>
13. <https://www.linkedin.com/pulse/5-essential-design-patterns-building-collaborative-systems-prakash-cn5nc>
14. <https://deta.surf>
15. <https://www.sciencedirect.com/science/article/pii/S0268401224001014>
16. [https://www.reddit.com/r/diabrowser/comments/1l2lmb9/how\\_does\\_dia\\_compare\\_to\\_deta\\_surf/](https://www.reddit.com/r/diabrowser/comments/1l2lmb9/how_does_dia_compare_to_deta_surf/)
17. <https://asana.com/resources/what-is-collaborative-intelligence>
18. <https://www.youtube.com/watch?v=-FJf3qaVsCA>
19. <https://www.monitask.com/en/business-glossary/collaborative-intelligence>
20. <https://www.kdnuggets.com/collaborative-intelligence-maximizing-human-ai-partnerships-workplace>
21. <https://orq.ai/blog/why-do-multi-agent-llm-systems-fail>