Weekly Progress Report (October 13-20, 2025)

What did you do?

The focus was on improving the agent's ability to understand user queries and correctly route requests to appropriate API endpoints. I identified and resolved several critical issues with parameter extraction, API routing, and agent reasoning that were causing failures and incorrect responses:

- Debugging and fixing the BIE (Biodiversity Information Explorer) search functionality
- Implementing field validation to prevent unsupported filters from being sent to BIE API
- Enhancing temporal query parsing to handle phrases like "after 2020" and "between 2010-2020"
- Improving the system prompt to guide better tool selection and prevent infinite recursion
- Adding comprehensive error handling and logging throughout the agent workflow

What new capabilities does your agent have?

The agent now has several enhanced capabilities:

Smart API Routing: The agent can now intelligently choose between occurrence search (for records, sightings, temporal/spatial filters) and BIE search (for taxonomy, species profiles, metadata) based on user query intent.

Robust Temporal Filtering: Enhanced parameter extraction now correctly handles natural language time expressions:

Field Validation with Caching: Implemented a cached validation system using functools.cache that fetches valid BIE index fields and filters out unsupported parameters before API calls, preventing errors.

Improved Stop Conditions: Added clear reasoning in the system prompt to prevent infinite loops and unnecessary tool calls, ensuring the agent calls finish() after presenting requested information.

What problems are you facing?

Despite improvements, the agent occasionally still enters recursion loops when it doesn't recognize that a query has been successfully fulfilled, particularly with taxonomy queries where it expects more detailed hierarchical data than the BIE API provides.

Some queries result in asyncio.CancelledError exceptions, likely due to long-running API calls or timeouts in the orchestration framework.

The LLM sometimes still misses filters in complex multi-parameter queries, requires further prompt refinement and possibly fallback rule-based parsing.

What will you do next week?

Implement Comprehensive Error Handling: Add proper timeout handling, retry logic, and graceful degradation for API failures and async cancellation errors.

Enhance System Prompt Engineering: Further refine the system prompt with more specific examples and clearer stop conditions to eliminate remaining recursion issues.
Add Fallback Parameter Extraction: Implement rule-based temporal expression parsing as a fallback when LLM extraction fails, ensuring robust handling of time-based queries.