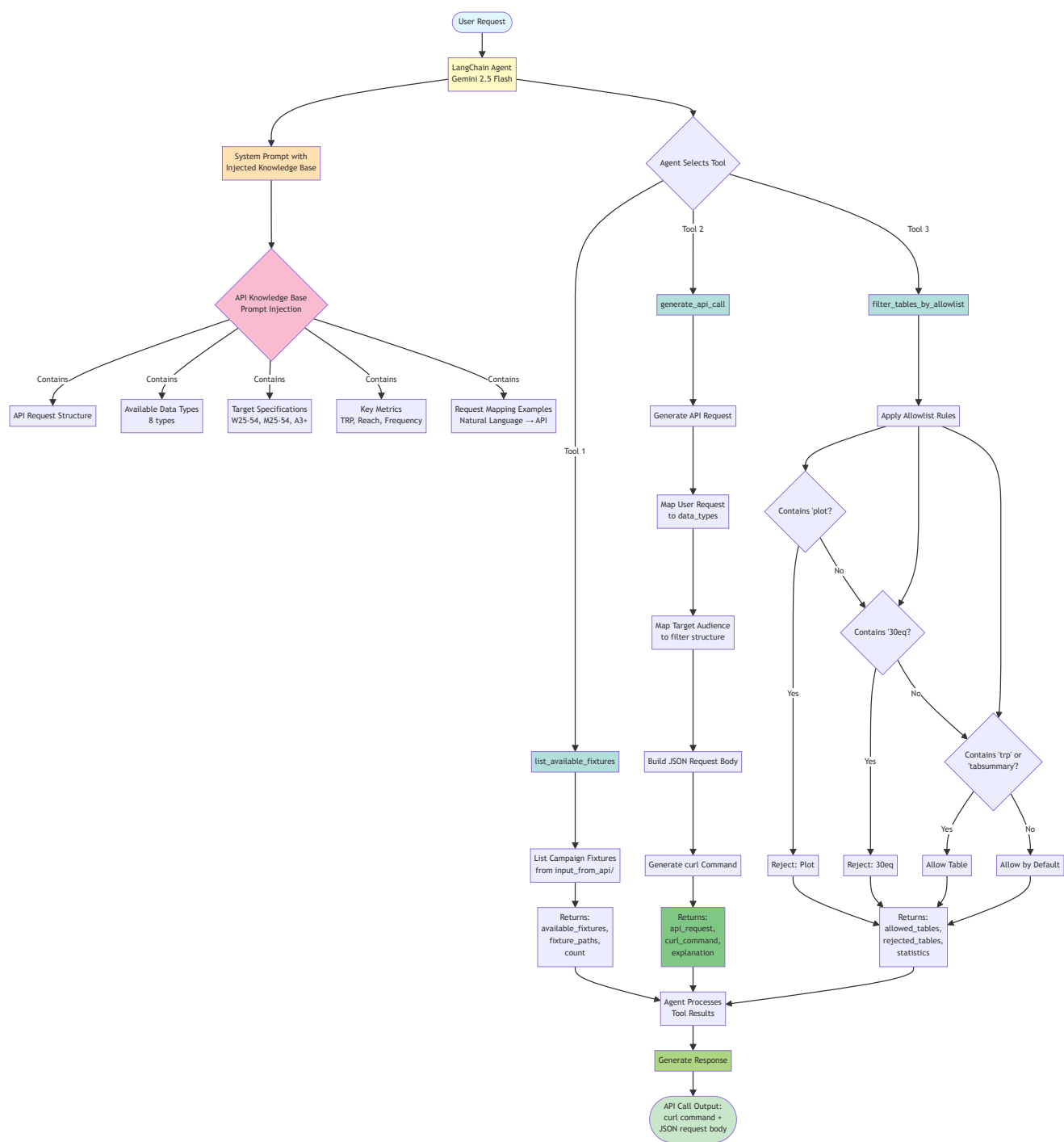


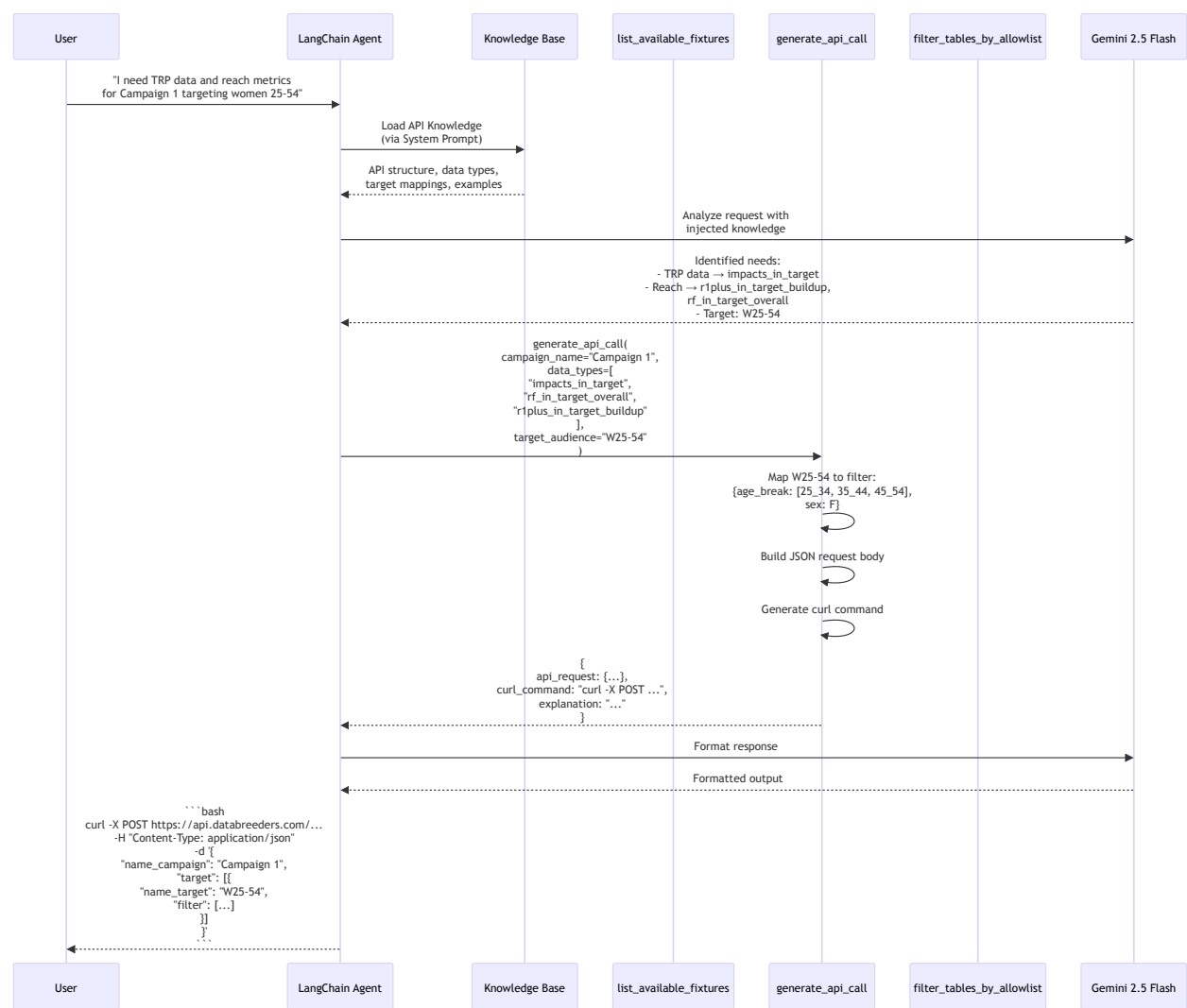
Lesson 1: Agent Architecture Flowchart

This flowchart illustrates the complete architecture of the dataBreeders API Input Analyzer Agent, showing how user requests are transformed into API calls through LangChain tools and knowledge injection.

Agent Architecture Overview

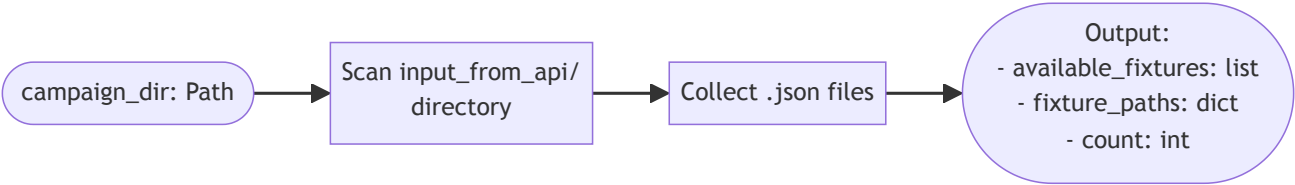


Detailed Flow: User Request to API Call



Tool Details

Tool 1: list_available_fixtures

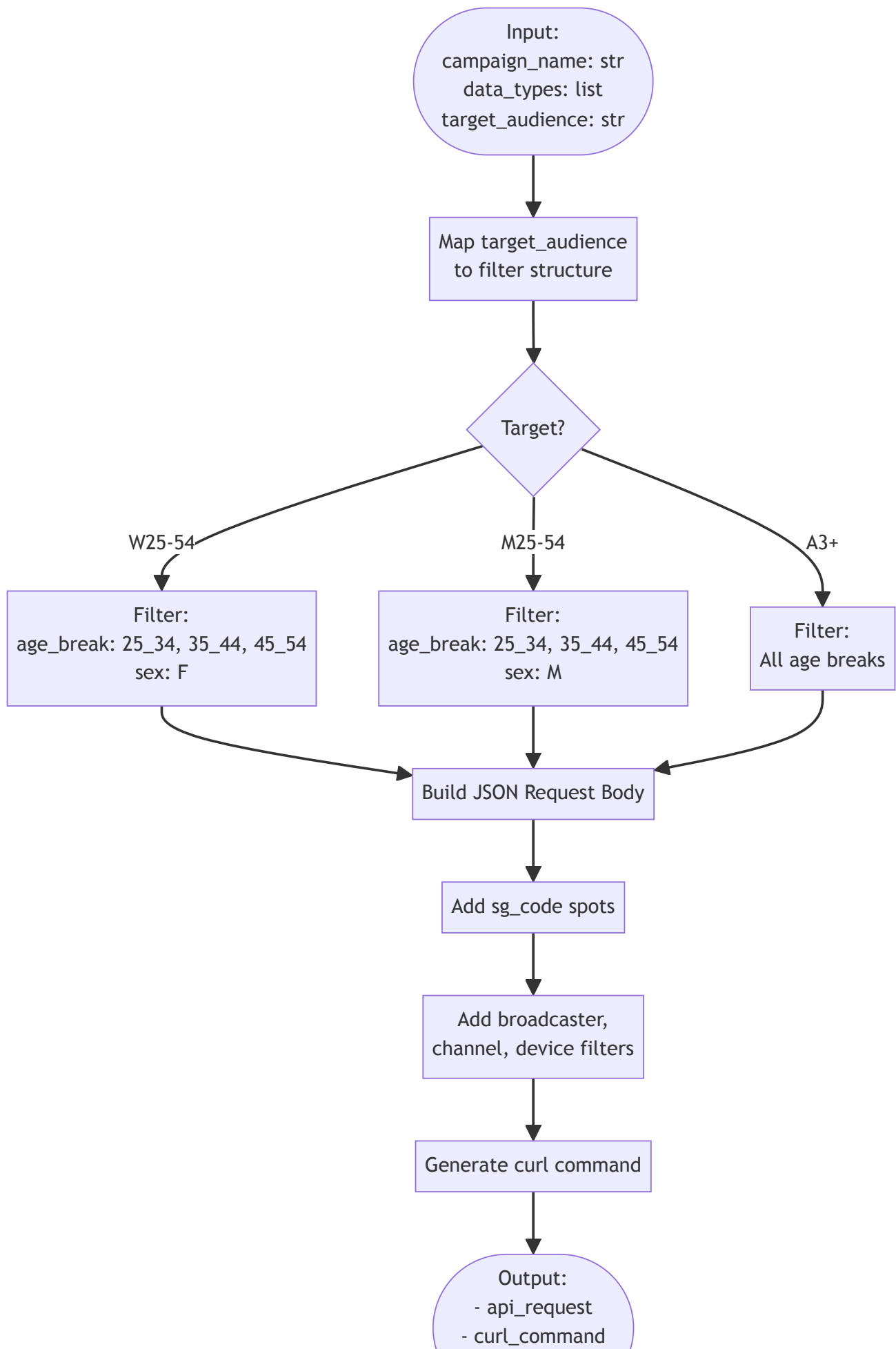


Purpose: Discover available data fixtures in Campaign 1 directory

Example Output:

```
{
  "campaign": "Campaign 1",
  "available_fixtures": [
    "impacts_by_sex_age",
    "impacts_in_target",
    "r1plus_in_target_buildup",
    "rf_in_target_overall",
    ...
  ],
  "count": 8
}
```

Tool 2: generate_api_call

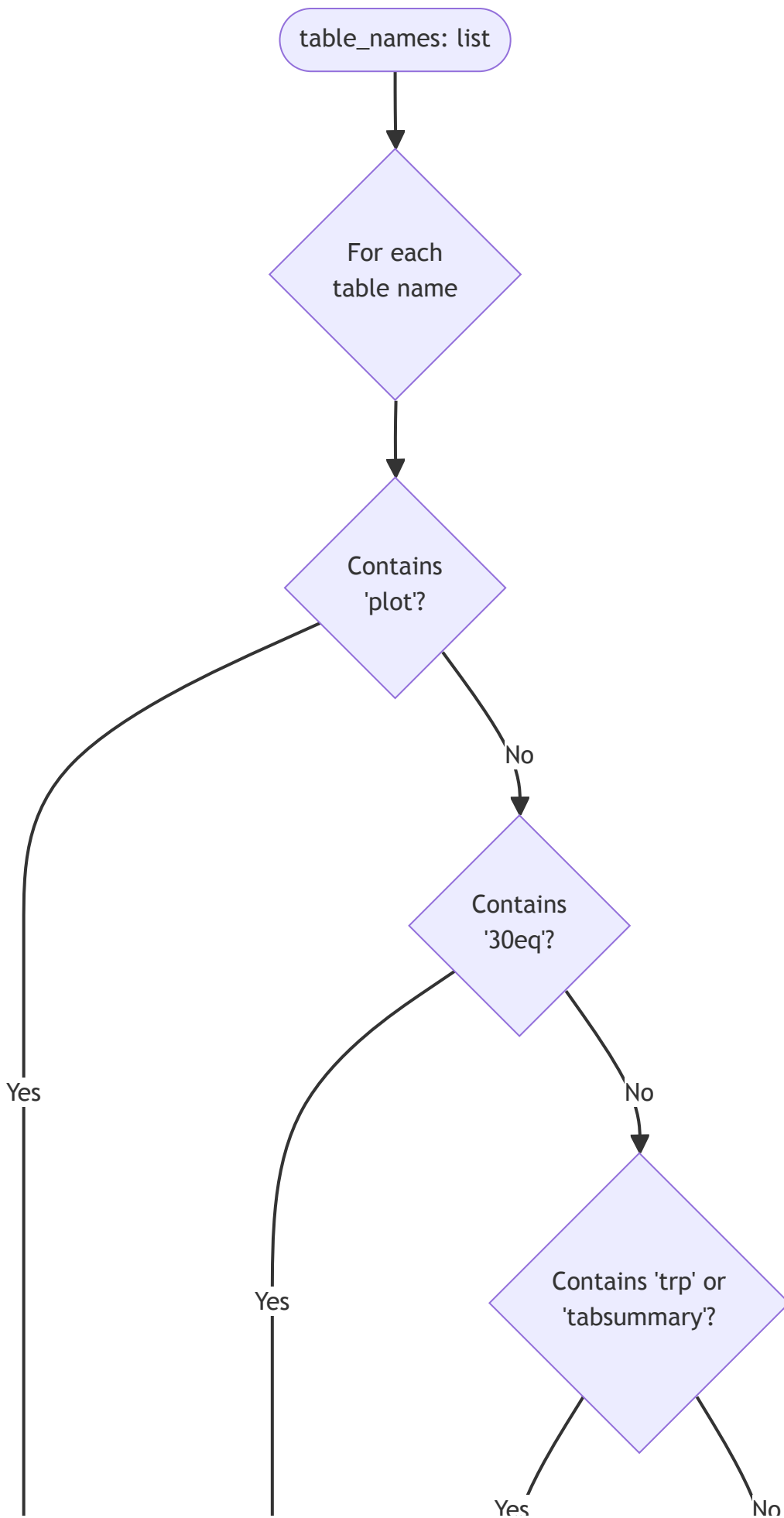


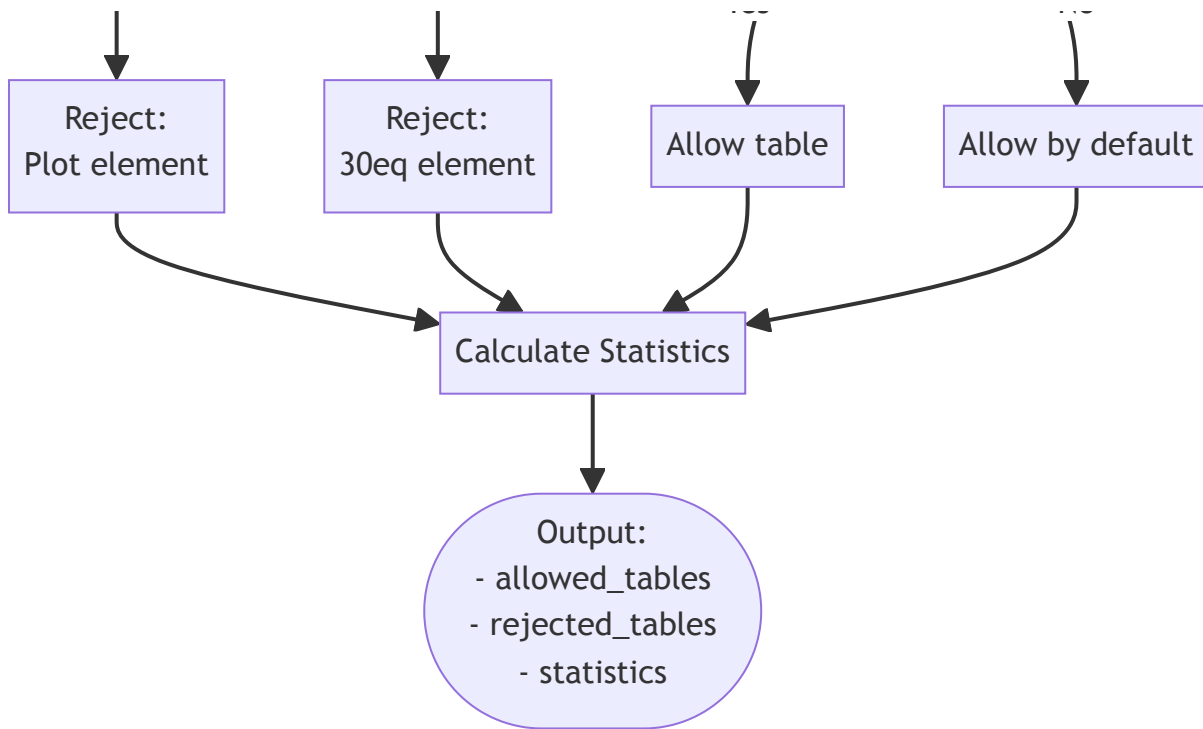
Purpose: Generate actual dataBreeders API call from natural language request

Example Output:

```
{
  "api_request": {
    "endpoint": "https://api.databreeders.com/v1/campaign/data",
    "method": "POST",
    "headers": {
      "Content-Type": "application/json",
      "Authorization": "Bearer YOUR_API_KEY"
    },
    "body": {
      "name_campaign": "Campaign 1",
      "target": [...],
      "sg_code": [...],
      "filter": {...}
    }
  },
  "curl_command": "curl -X POST ...",
  "explanation": "This API call will fetch..."
}
```

Tool 3: filter_tables_by_allowlist





Purpose: Filter table element keys based on AGENTS.md allowlist rules

Allowlist Rules:

- INCLUDE: TRP tables (contain 'trp')
- INCLUDE: TabSummary tables (contain 'tabsummary')
- EXCLUDE: Plot elements (contain 'plot') -- Just for testing purposes
- EXCLUDE: 30eq elements (contain '30eq') -- Just for testing purposes

Example Output:

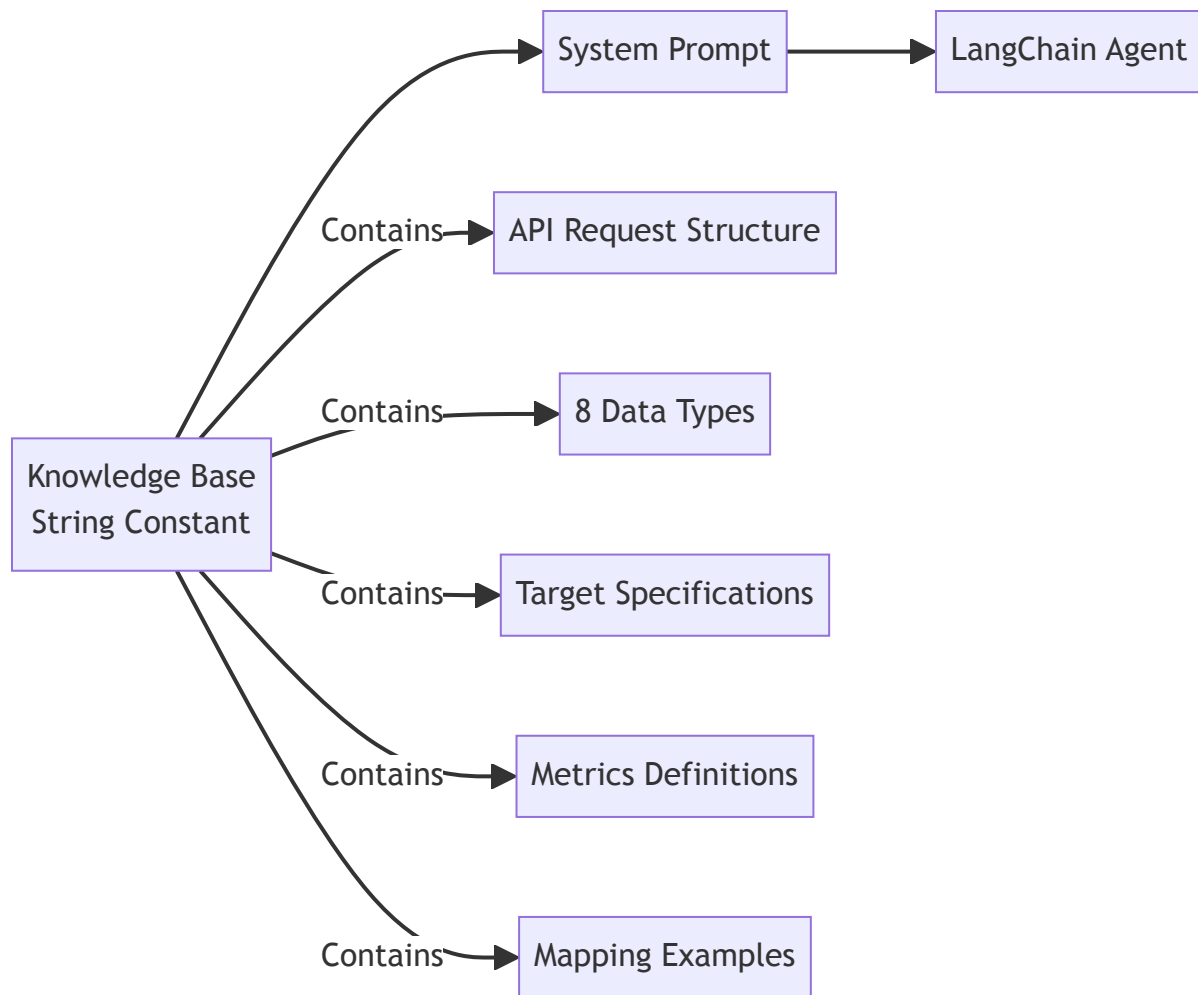
```

{
  "allowed_tables": [
    "standard_tabcontacts_table_contact_sexage_trp_raw",
    "standard_tabsummary_table_contactreach_target_perc"
  ],
  "rejected_tables": [
    {
      "name": "standard_tabcontacts_plot_contact_sexage_abs_raw",
      "reason": "excluded: contains 'plot'"
    }
  ],
  "statistics": {
    "total_input": 5,
    "allowed_count": 3,
    "rejected_count": 2
  }
}

```

Knowledge Base Injection

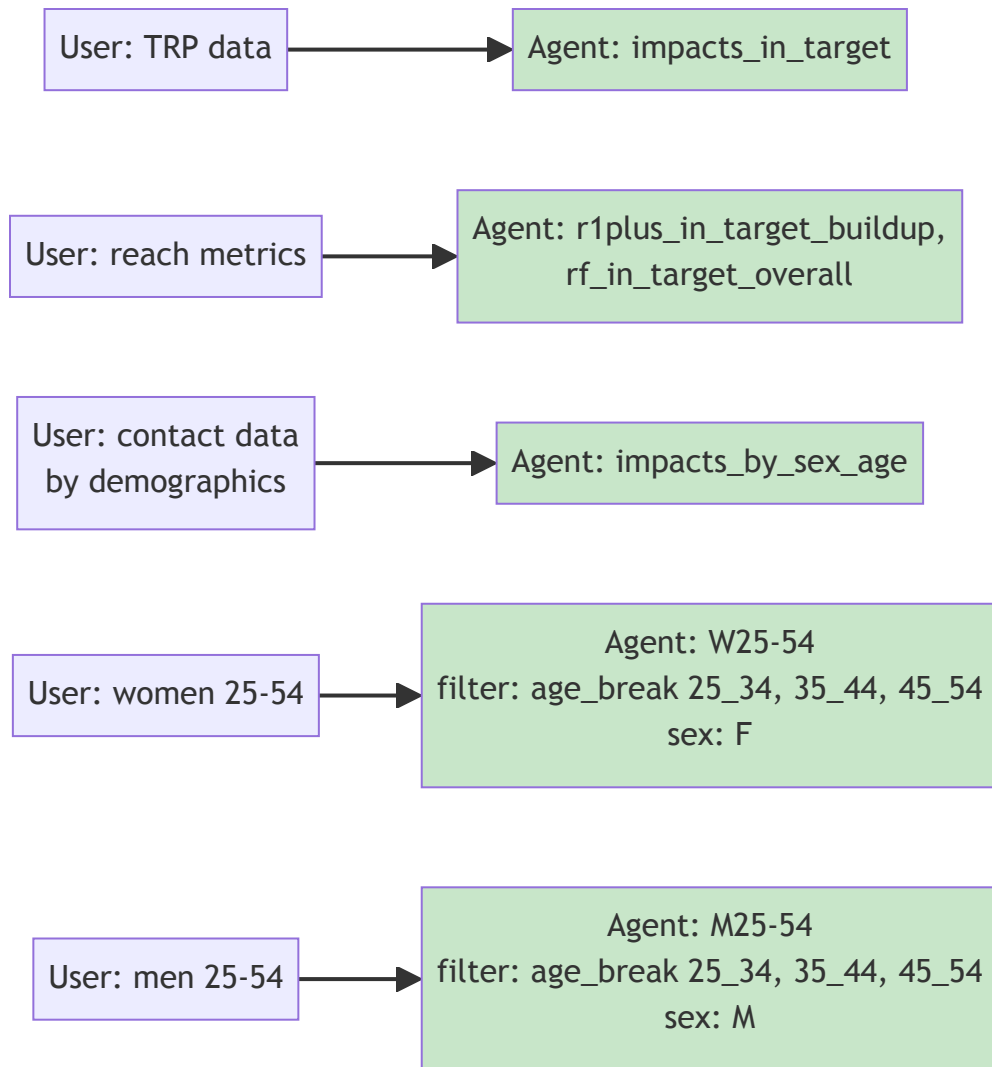
The agent uses **prompt injection** (not embeddings) to integrate dataBreeders API knowledge:



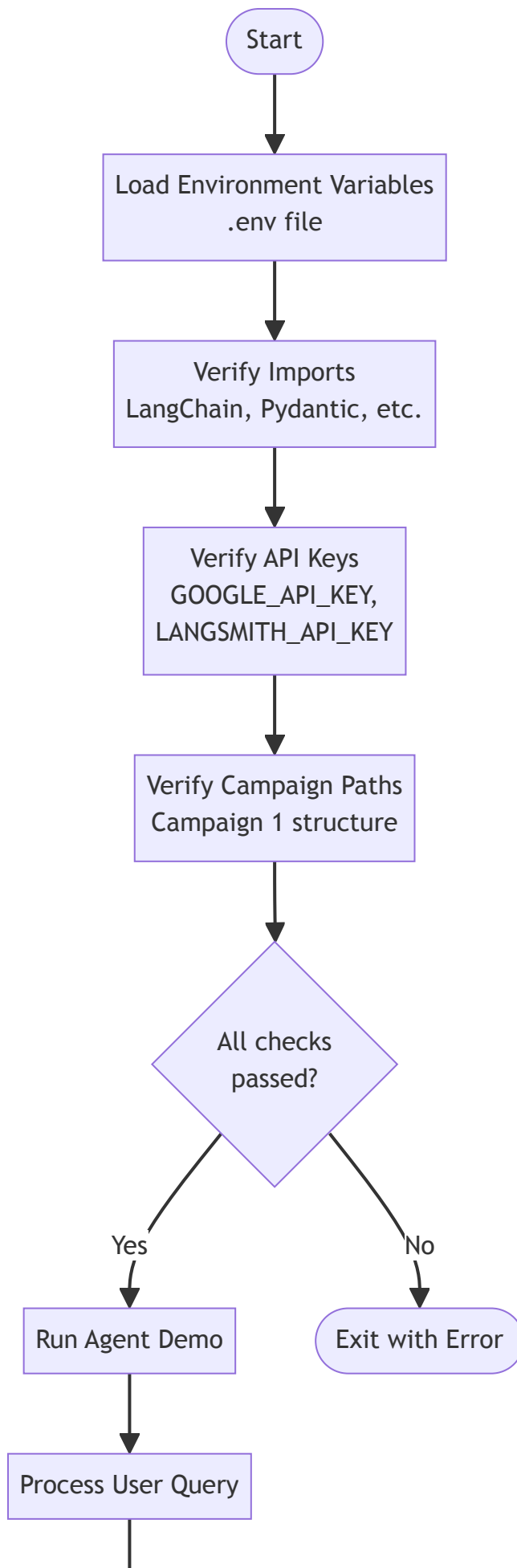
Advantages of Prompt Injection:

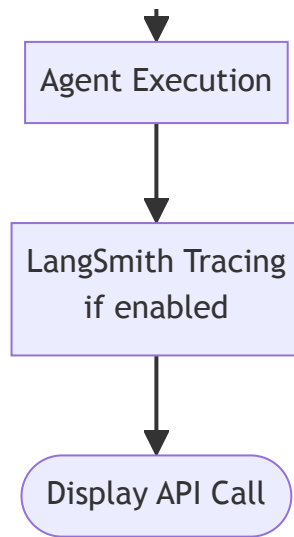
- Simpler (no vector database)
- Faster (no embedding generation)
- Deterministic (same knowledge always)
- Traceable (visible in LangSmith)
- Sufficient (~100 lines fits in context)

Request Transformation Examples



Environment Setup & Execution

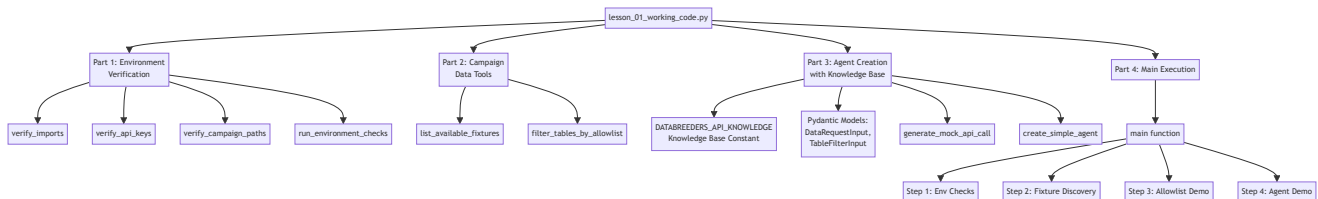




Key Design Principles

1. **Local-First**: Uses Campaign 1 fixtures, no external API calls
2. **Deterministic**: Temperature=0.0 for reproducible results
3. **Traceable**: LangSmith integration for debugging
4. **Allowlist-Driven**: Strict table filtering rules
5. **Knowledge Injection**: API knowledge in prompt, not embeddings
6. **API Call Generator**: Transforms requests to curl commands, not file searches

Files & Components



Running the Agent

```
# Basic execution
python docs/lessons/lesson_01_working_code.py
```

```
# With custom query
python docs/lessons/lesson_01_working_code.py --query "Get TRP data for women 25-54"
```



Expected Output

The agent will:

1. Verify environment setup
2. List available fixtures
3. Demonstrate table filtering
4. Transform user request into API call
5. Return curl command + JSON request body