MCP WORKSHOP

ISHITA GUPTA

WHY DO WE NEED MCP?

You are a busy student who tells people you want to get lunch, but never actually follows up

WHY DO WE NEED MCP?

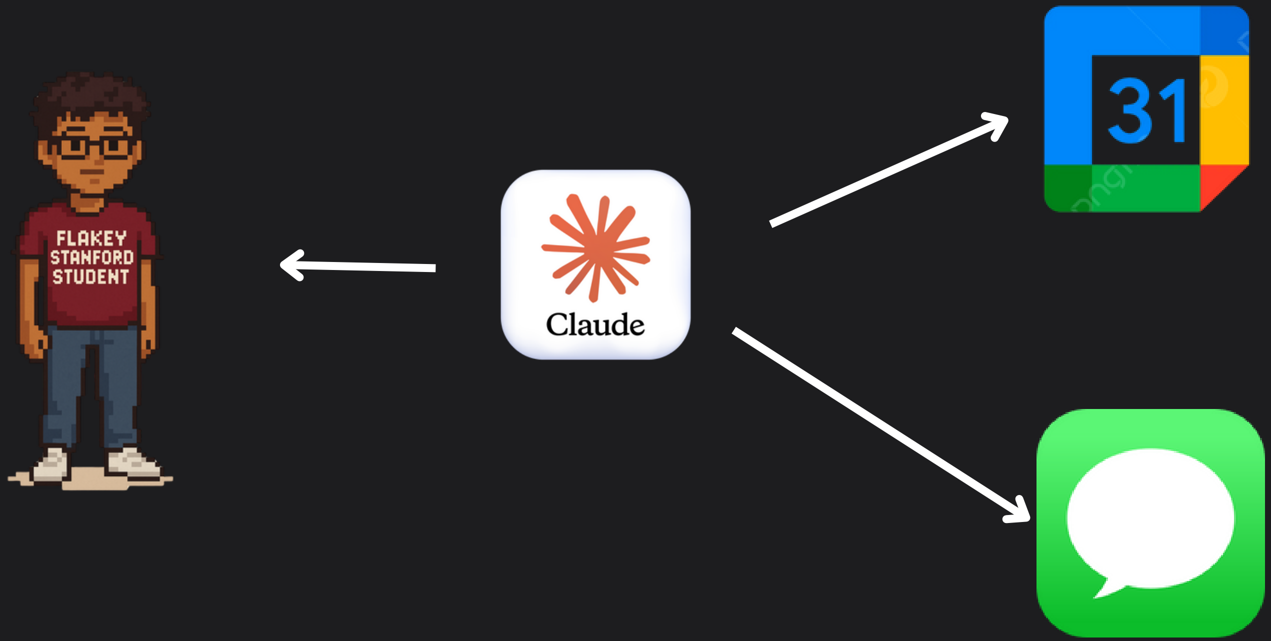
You want to build a simple agent that helps with this problem

WHY DO WE NEED MCP?

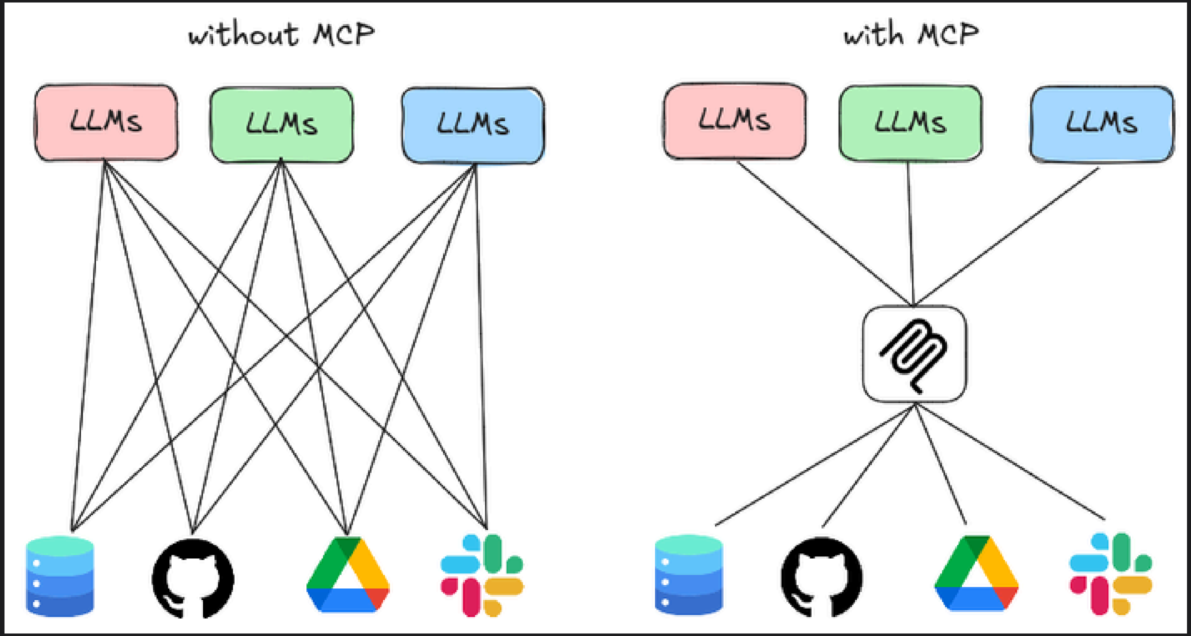
You want to build a simple agent that

1. Checks you calender
2. Texts your friend some times you are free to get lunch

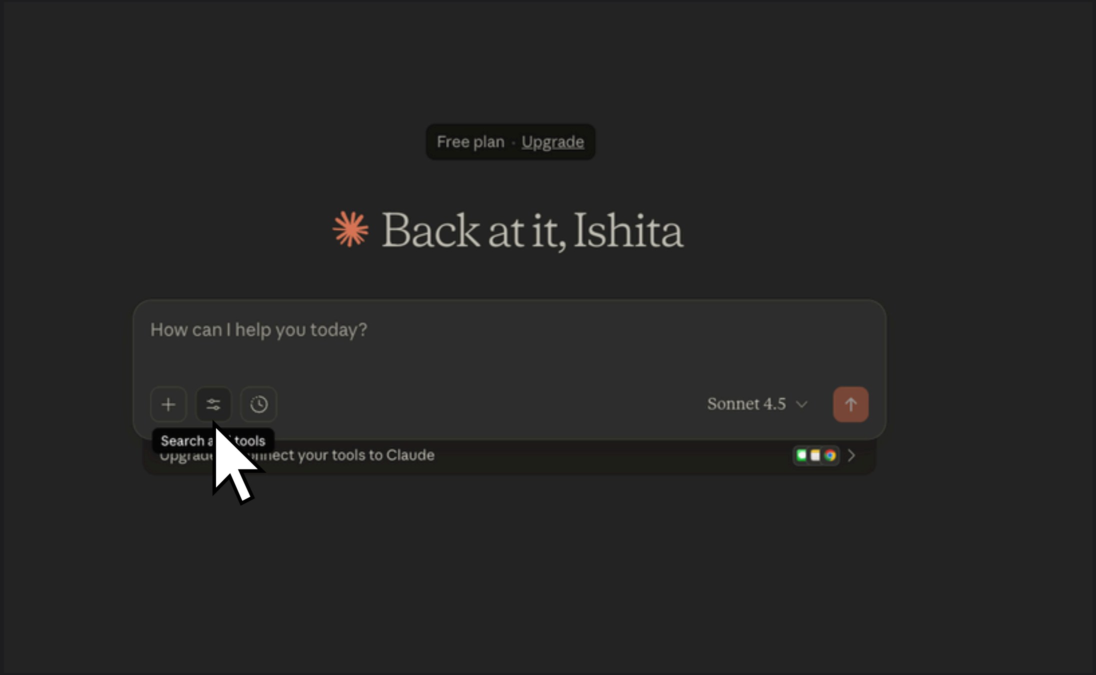
WHY DO WE NEED MCP?



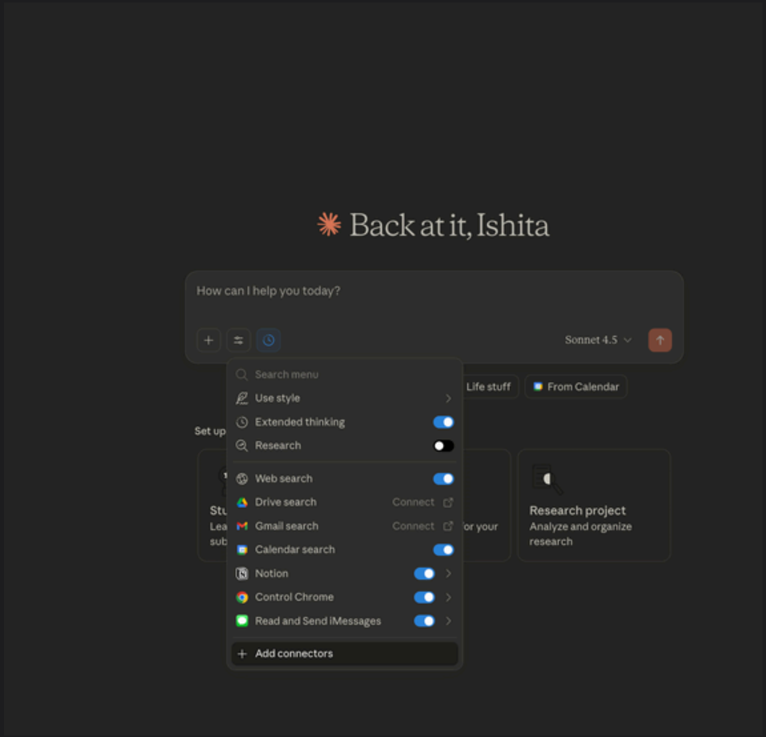
WHY DO WE NEED MCP?



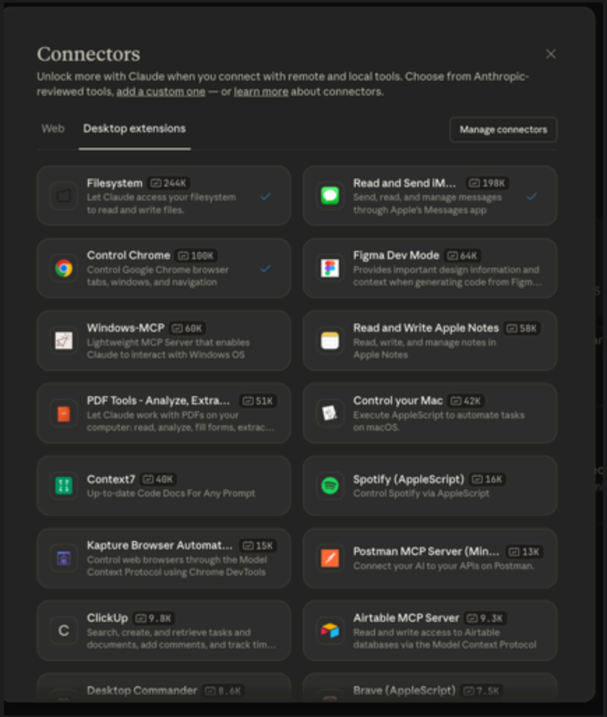
LETS MAKE AGENT!



LETS MAKE AGENT!



LETS MAKE AGENT!



Connectors

Unlock more with Claude when you connect with remote and local tools. Choose from Anthropic-reviewed tools, add a custom one — or learn more about connectors.

Filesystem 244K ✓
Let Claude access your filesystem to read and write files.

Read and Send iM... 198K ✓
Send, read, and manage messages through Apple's Messages app

Control Chrome 188K ✓
Control Google Chrome browser tabs, windows, and navigation

Figma Dev Mode 64K
Provides important design information and context when generating code from Figm...

Windows-MCP 68K
Lightweight MCP Server that enables Claude to interact with Windows OS

Read and Write Apple Notes 58K
Read, write, and manage notes in Apple Notes

PDF Tools - Analyze, Extra... 51K
Let Claude work with PDFs on your computer: read, analyze, fill forms, extrac...

Control your Mac 42K
Execute AppleScript to automate tasks on macOS.

Context7 48K
Up-to-date Code Docs For Any Prompt

Spotify (AppleScript) 16K
Control Spotify via AppleScript

Kapture Browser Automat... 15K
Control web browsers through the Model Context Protocol using Chrome DevTools

Postman MCP Server (Min... 13K
Connect your AI to your APIs on Postman.

ClickUp 9.8K
Search, create, and retrieve tasks and documents, add comments, and track tim...

Airtable MCP Server 9.3K
Read and write access to Airtable databases via the Model Context Protocol

Desktop Commander 8.6K

Brave (AppleScript) 7.5K

AGENT ACTIVITY 🙂

MCP ISN’T MAGIC:

- When you asked the agent to send a message, what do you think it did first — before touching any tool?
- How do you think it decided which tool could help it complete that request?
- Once it picked a tool, what information do you think it had to pass along for the tool to actually work?

HOW IT WORKS

Tool Selection — choosing which external capability to use.

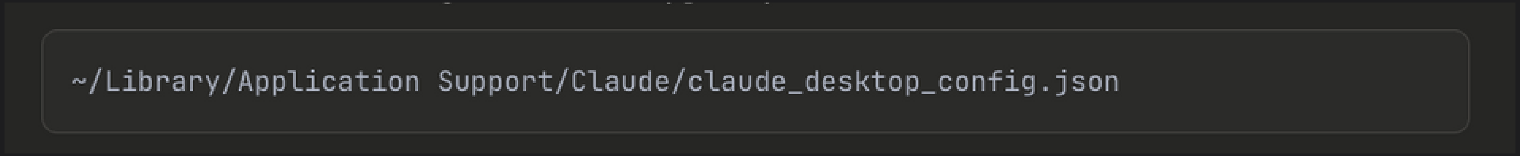
Tool Invocation — the actual call and context that goes out.

HOW IT WORKS

Tool Description — how an agent knows what each tool does.

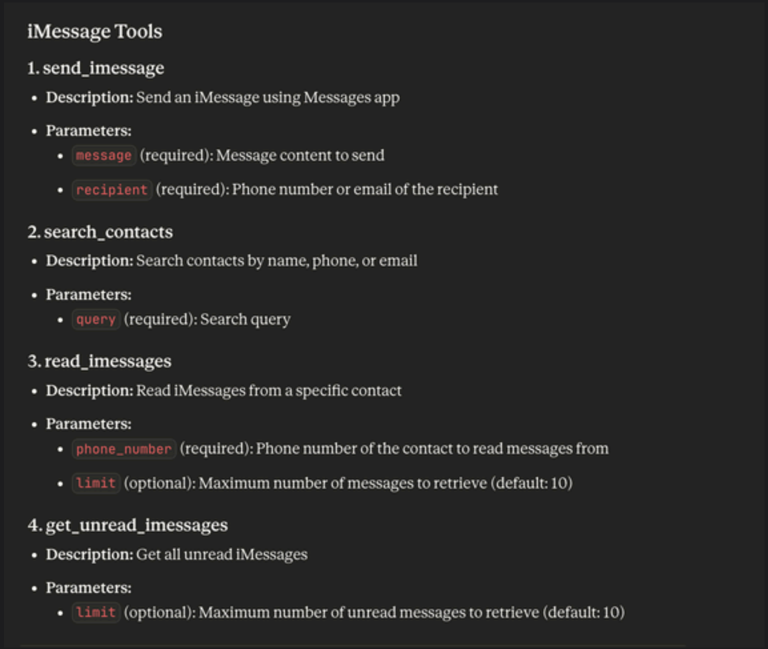
Tool Calling — what parameters to pass into the tool

WHERE ITS STORED



```
~/Library/Application Support/Claude/claude_desktop_config.json
```

IMESSAGE TOOLS



iMessage Tools

1. send_imessage

- **Description:** Send an iMessage using Messages app
- **Parameters:**
    - `message` (required): Message content to send
    - `recipient` (required): Phone number or email of the recipient

2. search_contacts

- **Description:** Search contacts by name, phone, or email
- **Parameters:**
    - `query` (required): Search query

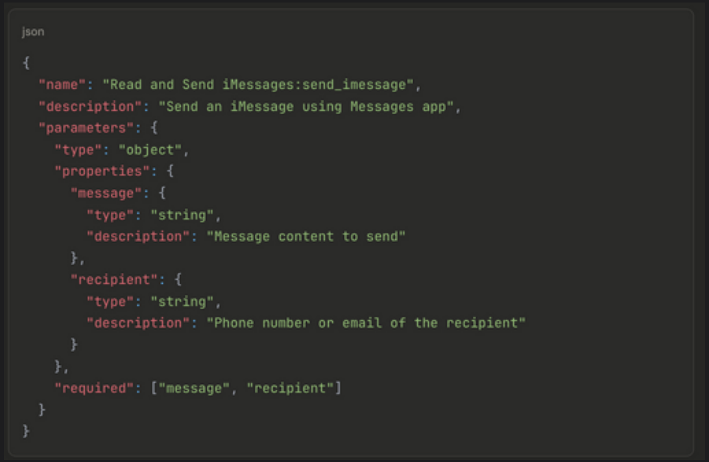
3. read_imessages

- **Description:** Read iMessages from a specific contact
- **Parameters:**
    - `phone_number` (required): Phone number of the contact to read messages from
    - `limit` (optional): Maximum number of messages to retrieve (default: 10)

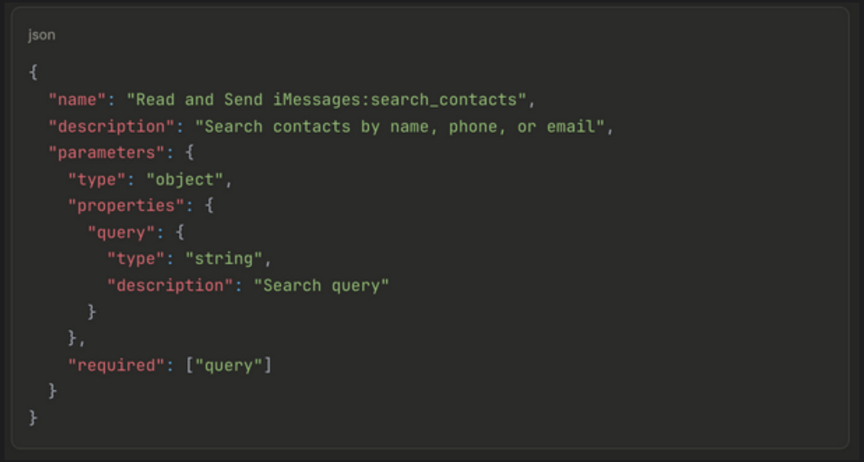
4. get_unread_imessages

- **Description:** Get all unread iMessages
- **Parameters:**
    - `limit` (optional): Maximum number of unread messages to retrieve (default: 10)

IMESSAGE TOOLS

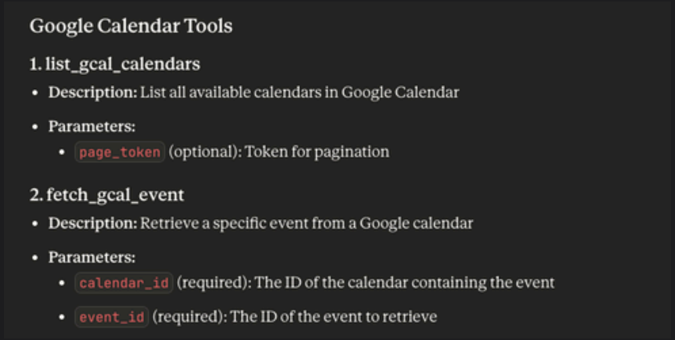


```json
{
  "name": "Read and Send iMessages:send_imessage",
  "description": "Send an iMessage using Messages app",
  "parameters": {
    "type": "object",
    "properties": {
      "message": {
        "type": "string",
        "description": "Message content to send"
      },
      "recipient": {
        "type": "string",
        "description": "Phone number or email of the recipient"
      }
    },
    "required": ["message", "recipient"]
  }
}
```



```json
{
  "name": "Read and Send iMessages:search_contacts",
  "description": "Search contacts by name, phone, or email",
  "parameters": {
    "type": "object",
    "properties": {
      "query": {
        "type": "string",
        "description": "Search query"
      }
    },
    "required": ["query"]
  }
}
```

GOOGLE CALENDAR TOOLS



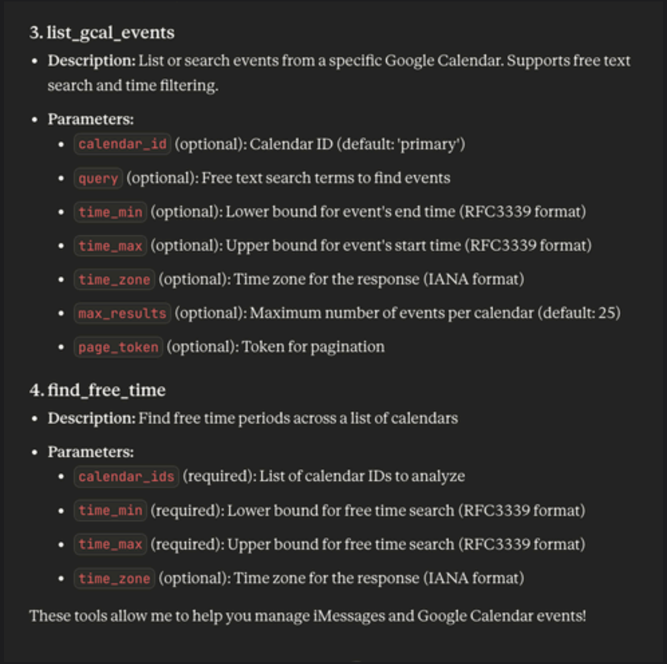
Google Calendar Tools

1. list_gcal_calendars

- **Description:** List all available calendars in Google Calendar
- **Parameters:**
  - `page_token` (optional): Token for pagination

2. fetch_gcal_event

- **Description:** Retrieve a specific event from a Google calendar
- **Parameters:**
  - `calendar_id` (required): The ID of the calendar containing the event
  - `event_id` (required): The ID of the event to retrieve



3. list_gcal_events

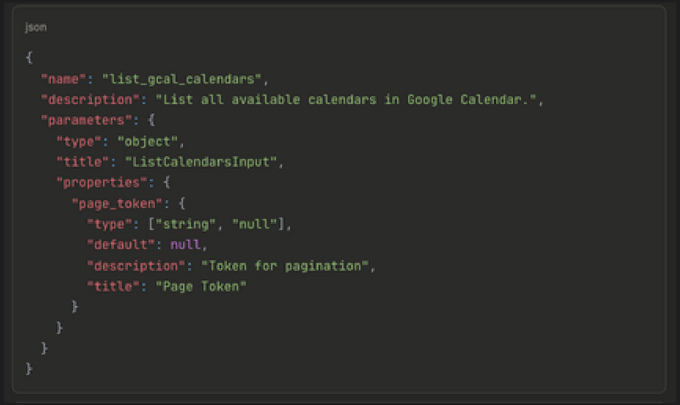
- **Description:** List or search events from a specific Google Calendar. Supports free text search and time filtering.
- **Parameters:**
  - `calendar_id` (optional): Calendar ID (default: 'primary')
  - `query` (optional): Free text search terms to find events
  - `time_min` (optional): Lower bound for event's end time (RFC3339 format)
  - `time_max` (optional): Upper bound for event's start time (RFC3339 format)
  - `time_zone` (optional): Time zone for the response (IANA format)
  - `max_results` (optional): Maximum number of events per calendar (default: 25)
  - `page_token` (optional): Token for pagination

4. find_free_time

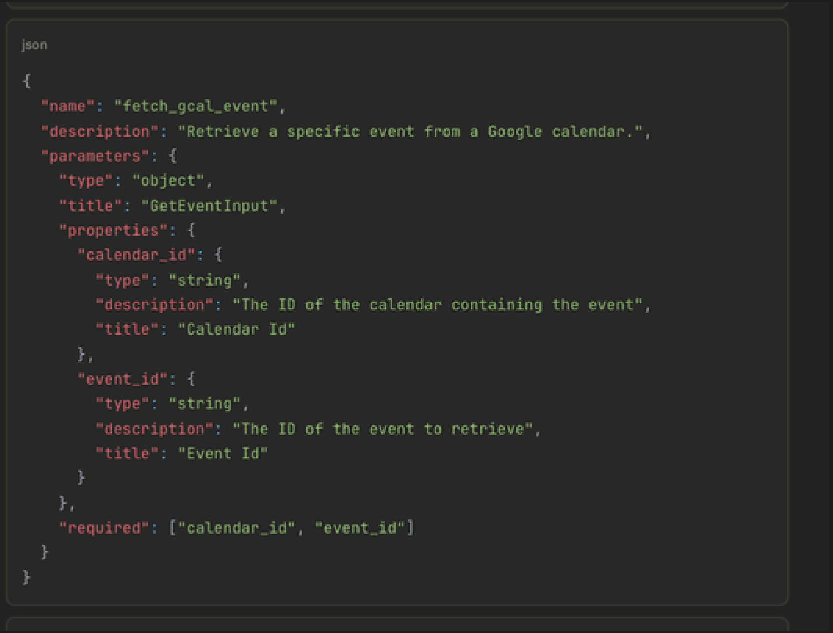
- **Description:** Find free time periods across a list of calendars
- **Parameters:**
  - `calendar_ids` (required): List of calendar IDs to analyze
  - `time_min` (required): Lower bound for free time search (RFC3339 format)
  - `time_max` (required): Upper bound for free time search (RFC3339 format)
  - `time_zone` (optional): Time zone for the response (IANA format)

These tools allow me to help you manage iMessages and Google Calendar events!

GOOGLE CALENDAR TOOLS



```json
{
  "name": "list_gcal_calendars",
  "description": "List all available calendars in Google Calendar.",
  "parameters": {
    "type": "object",
    "title": "ListCalendarsInput",
    "properties": {
      "page_token": {
        "type": ["string", "null"],
        "default": null,
        "description": "Token for pagination",
        "title": "Page Token"
      }
    }
  }
}
```



```json
{
  "name": "fetch_gcal_event",
  "description": "Retrieve a specific event from a Google calendar.",
  "parameters": {
    "type": "object",
    "title": "GetEventInput",
    "properties": {
      "calendar_id": {
        "type": "string",
        "description": "The ID of the calendar containing the event",
        "title": "Calendar Id"
      },
      "event_id": {
        "type": "string",
        "description": "The ID of the event to retrieve",
        "title": "Event Id"
      }
    },
    "required": ["calendar_id", "event_id"]
  }
}
```

WHAT DID I DO THIS SUMMER?

MCP Evals!

Built a Big Agent System

Tried to See if it did ok?

WHAT DID I DO THIS SUMMER?

WHAT DID I DO THIS SUMMER?

MCP

Evals!

WHAT DID I DO THIS SUMMER?

MCP
Evals!

WHAT DID I DO THIS SUMMER?

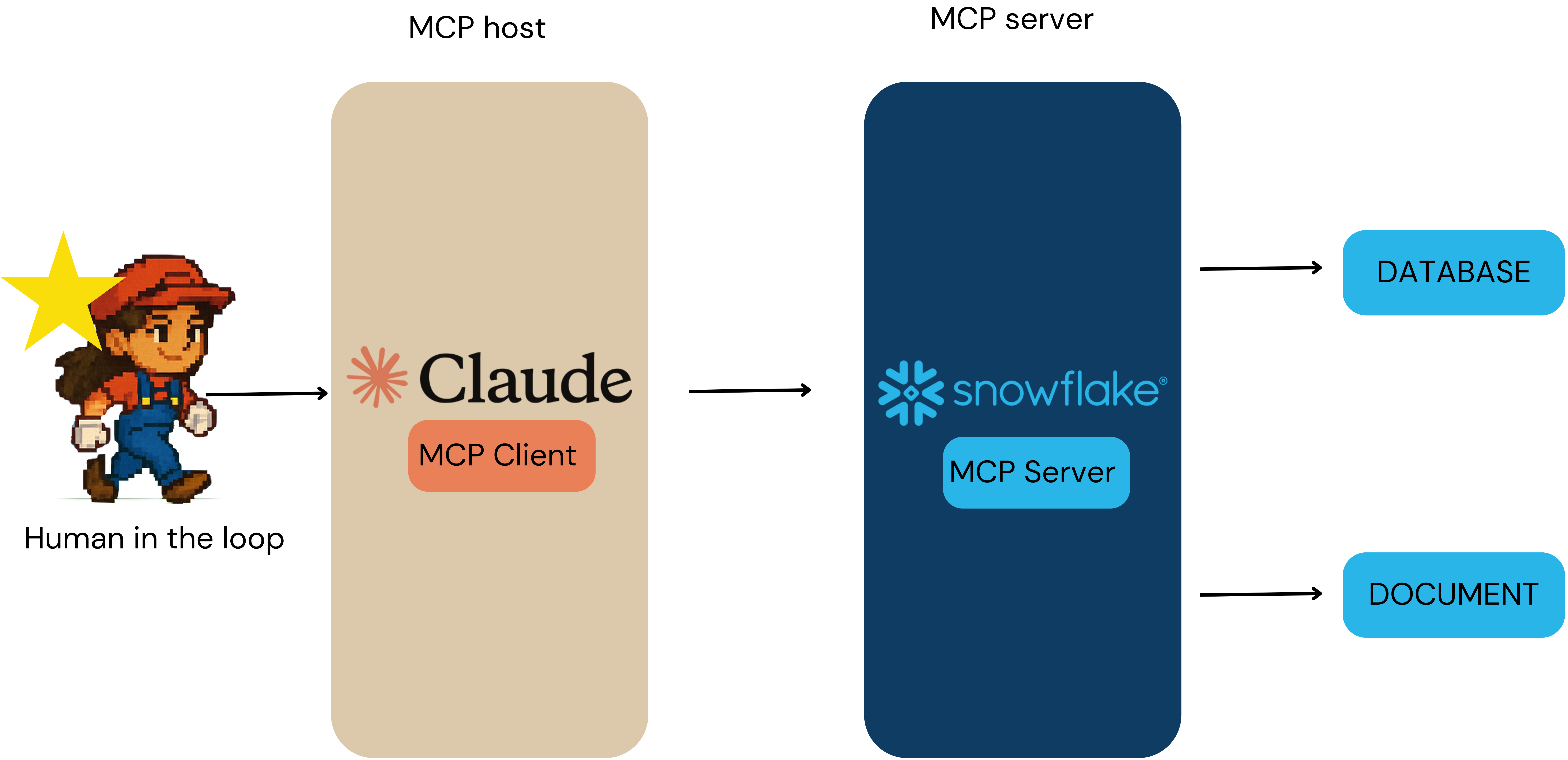
MCP
Evals!

WHAT DID I DO THIS SUMMER?

MCP Evals!

WHAT DID I DO THIS SUMMER?

MCP
Evals!



MCP host

MCP server

Human in the loop

Claude

MCP Client

snowflake

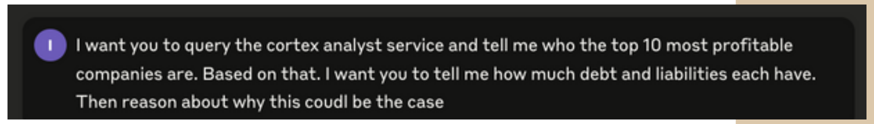
MCP Server

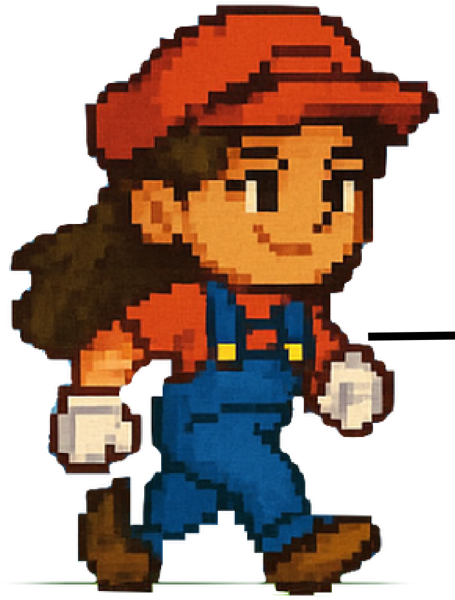
DATABASE

DOCUMENT

MCP host

MCP server

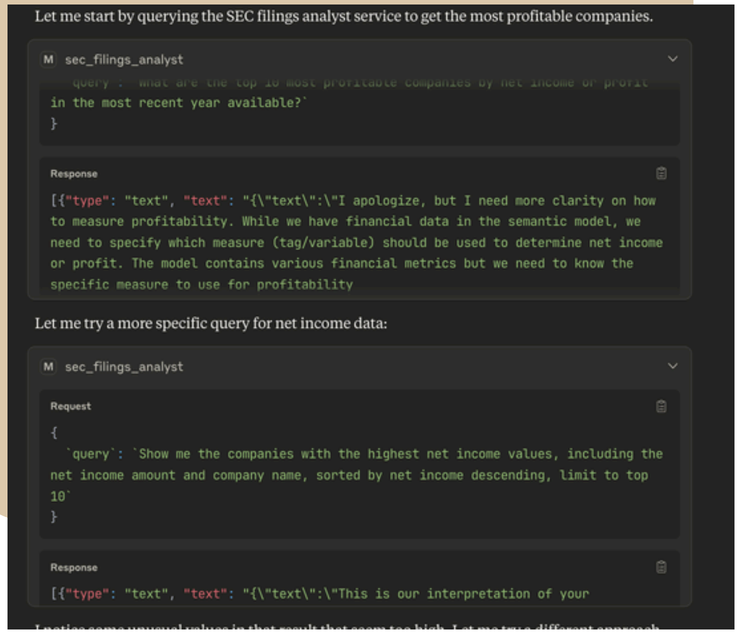

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case



Human in the loop

Claude

MCP Client


Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
  `query`: `What are the top 10 most profitable companies by net income or profit in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability
```

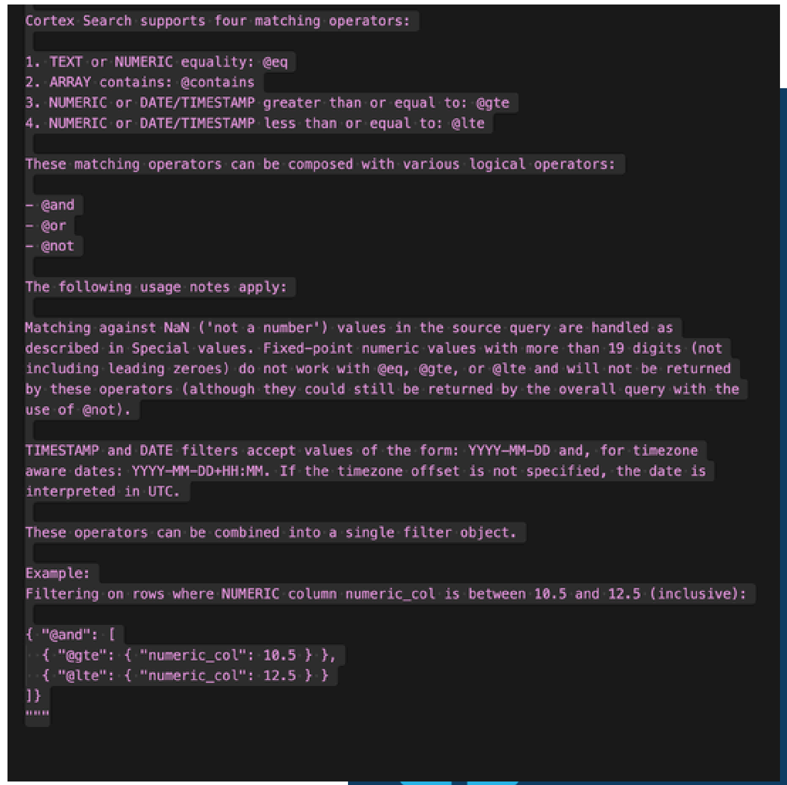
Let me try a more specific query for net income data:

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10`
}
```

Response

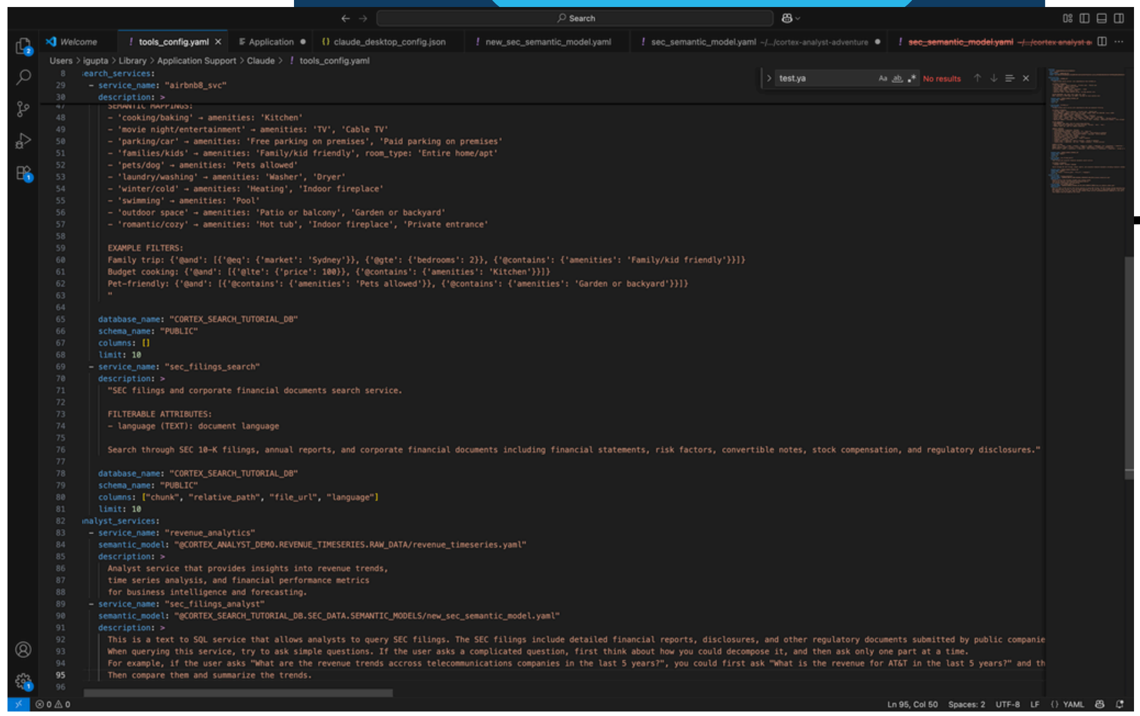
```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```



```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

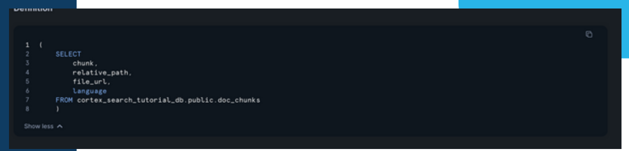
snowflake®

MCP Server

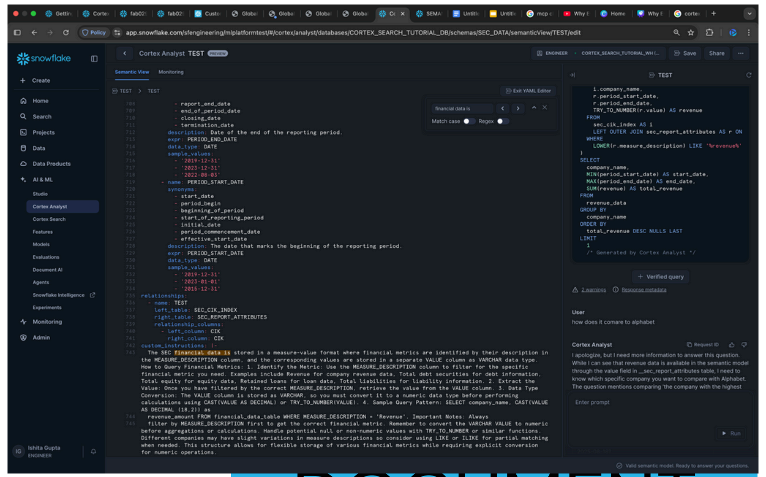

Search
Welcome
tools_config.yaml
Application
claude_desktop_config.json
new_sec_semantic_model.yaml
sec_semantic_model.yaml ~/.../cortex-analyst-adventure
sec_semantic_model.yaml
Users > igupta > Library > Application Support > Claude > tools_config.yaml
test.ya
No results
Ln 95, Col 50
Spaces: 2
UTF-8
LF
YAML

```
earch_services:
  - service_name: "airbnb8_svc"
    description: >
      SEMANTIC MAPPINGS:
      - 'cooking/baking' → amenities: 'Kitchen'
      - 'movie night/entertainment' → amenities: 'TV', 'Cable TV'
      - 'parking/car' → amenities: 'Free parking on premises', 'Paid parking on premises'
      - 'families/kids' → amenities: 'Family/kid friendly', room_type: 'Entire home/apt'
      - 'pets/dog' → amenities: 'Pets allowed'
      - 'laundry/washing' → amenities: 'Washer', 'Dryer'
      - 'winter/cold' → amenities: 'Heating', 'Indoor fireplace'
      - 'swimming' → amenities: 'Pool'
      - 'outdoor space' → amenities: 'Patio or balcony', 'Garden or backyard'
      - 'romantic/cozy' → amenities: 'Hot tub', 'Indoor fireplace', 'Private entrance'

      EXAMPLE FILTERS:
      Family trip: {'@and': [{'@eq': {'market': 'Sydney'}}, {'@gte': {'bedrooms': 2}}, {'@contains': {'amenities': 'Family/kid friendly'}}]}
      Budget cooking: {'@and': [{'@lte': {'price': 100}}, {'@contains': {'amenities': 'Kitchen'}}]}
      Pet-friendly: {'@and': [{'@contains': {'amenities': 'Pets allowed'}}, {'@contains': {'amenities': 'Garden or backyard'}}]}
      "

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: []
    limit: 10
  - service_name: "sec_filings_search"
    description: >
      "SEC filings and corporate financial documents search service.

      FILTERABLE ATTRIBUTES:
      - language (TEXT): document language

      Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: ["chunk", "relative_path", "file_url", "language"]
    limit: 10
nalyst_services:
  - service_name: "revenue_analytics"
    semantic_model: "@CORTEX_ANALYST_DEMO.REVENUE_TIMESERIES.RAW_DATA/revenue_timeseries.yaml"
    description: >
      Analyst service that provides insights into revenue trends,
      time series analysis, and financial performance metrics
      for business intelligence and forecasting.
  - service_name: "sec_filings_analyst"
    semantic_model: "@CORTEX_SEARCH_TUTORIAL_DB.SEC_DATA.SEMANTIC_MODELS/new_sec_semantic_model.yaml"
    description: >
      This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companie
      When querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time.
      For example, if the user asks "What are the revenue trends accross telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and th
      Then compare them and summarize the trends.
```

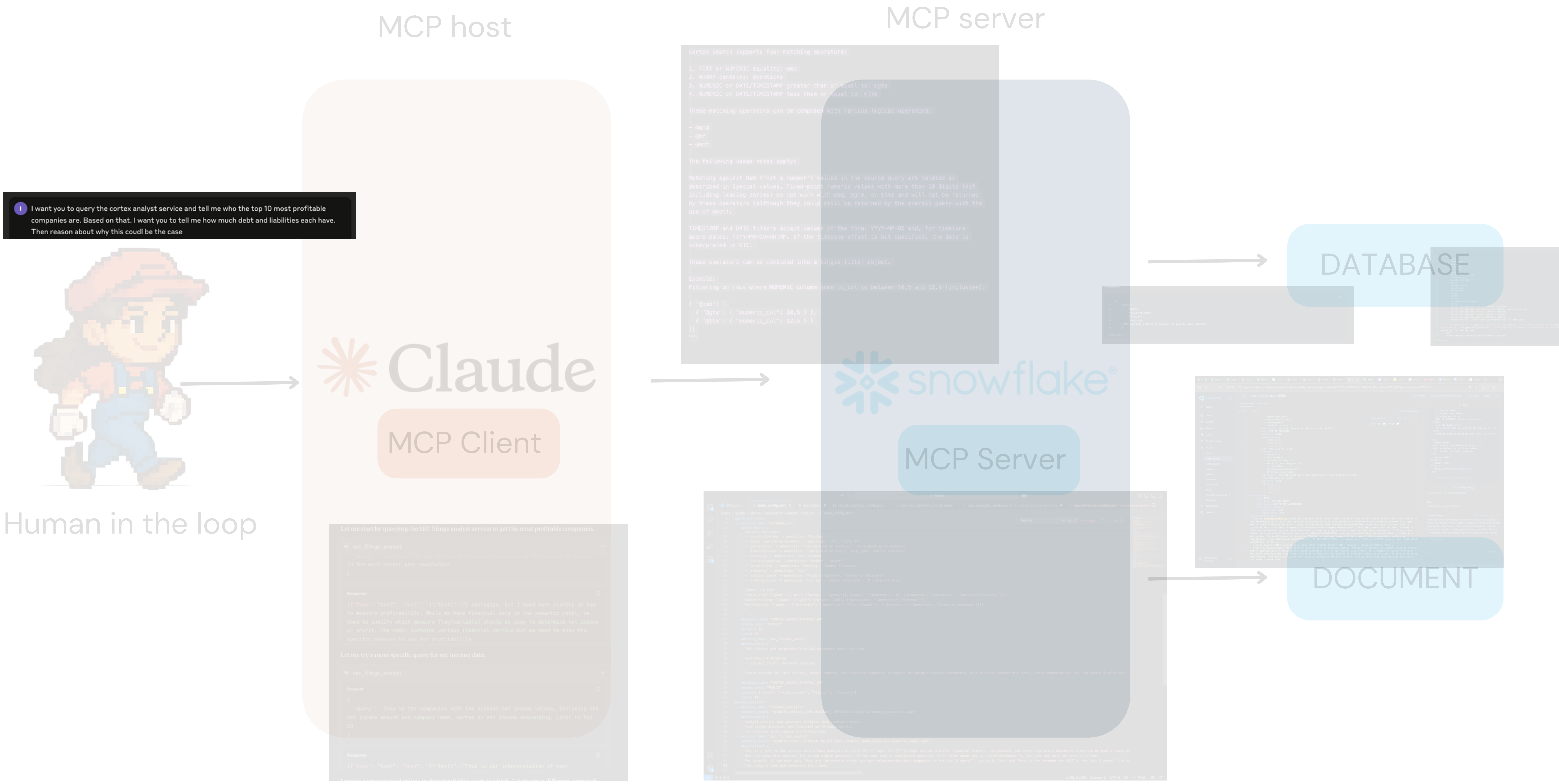
DATABA



```
(
    SELECT
        chunk,
        relative_path,
        file_url,
        language
    FROM cortex_search_tutorial_db.public.doc_chunks
    )
```



DOCUMENT



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

Let me start by querying the SEC filings analyst service to get the most profitable companies.

sec_filings_analyst

```
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

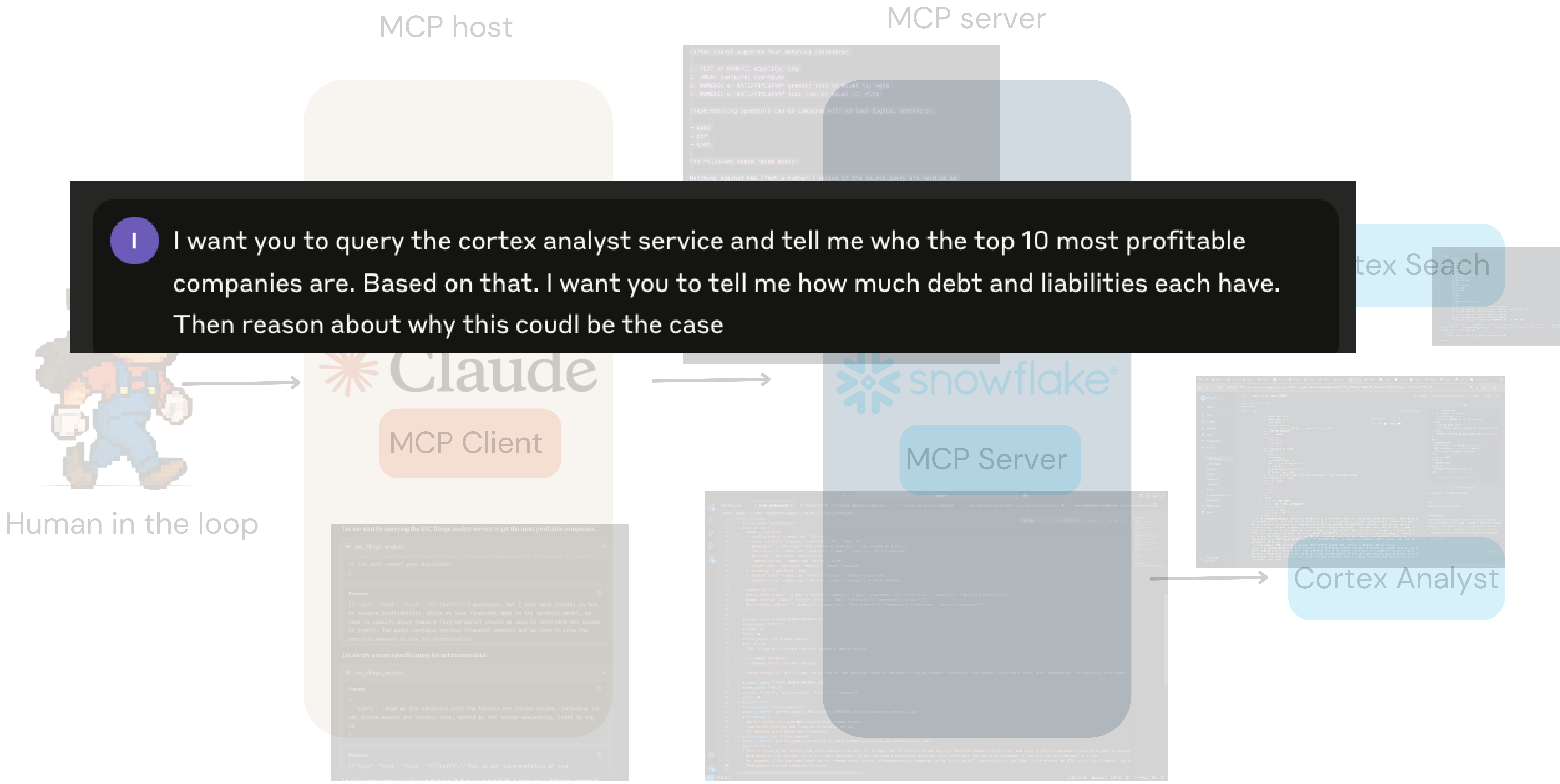
sec_filings_analyst

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```



MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

Human in the loop

MCP Client

MCP Server

tex Seach

Cortex Analyst

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
```

Let me start by querying the SEC filings analyst service to get the most profitable companies.

M sec_filings_analyst

```
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability
```

Let me try a more specific query for net income data:

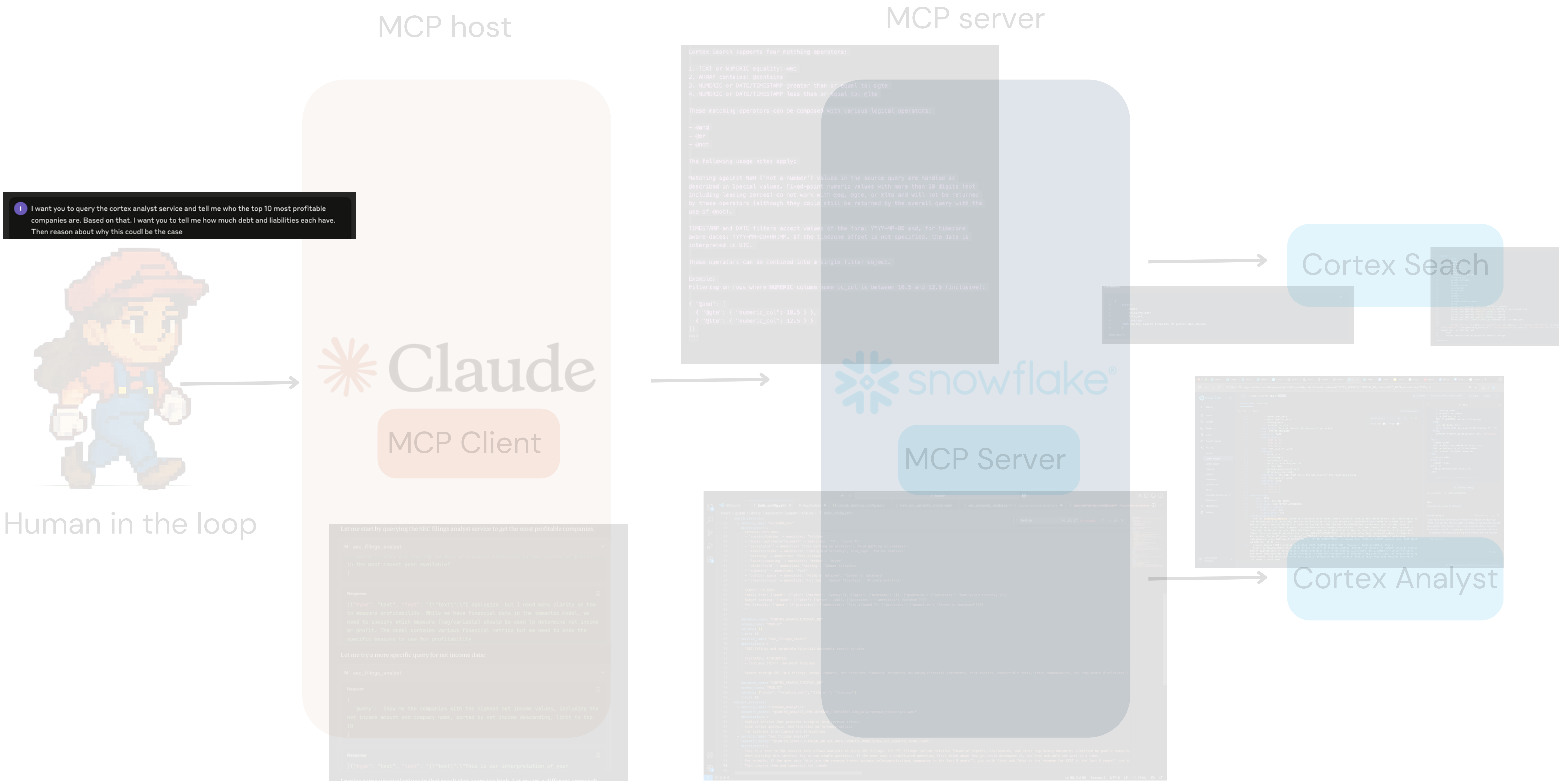
M sec_filings_analyst

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

Let me start by querying the SEC filings analyst service to get the most profitable companies.

sec_filings_analyst

```
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

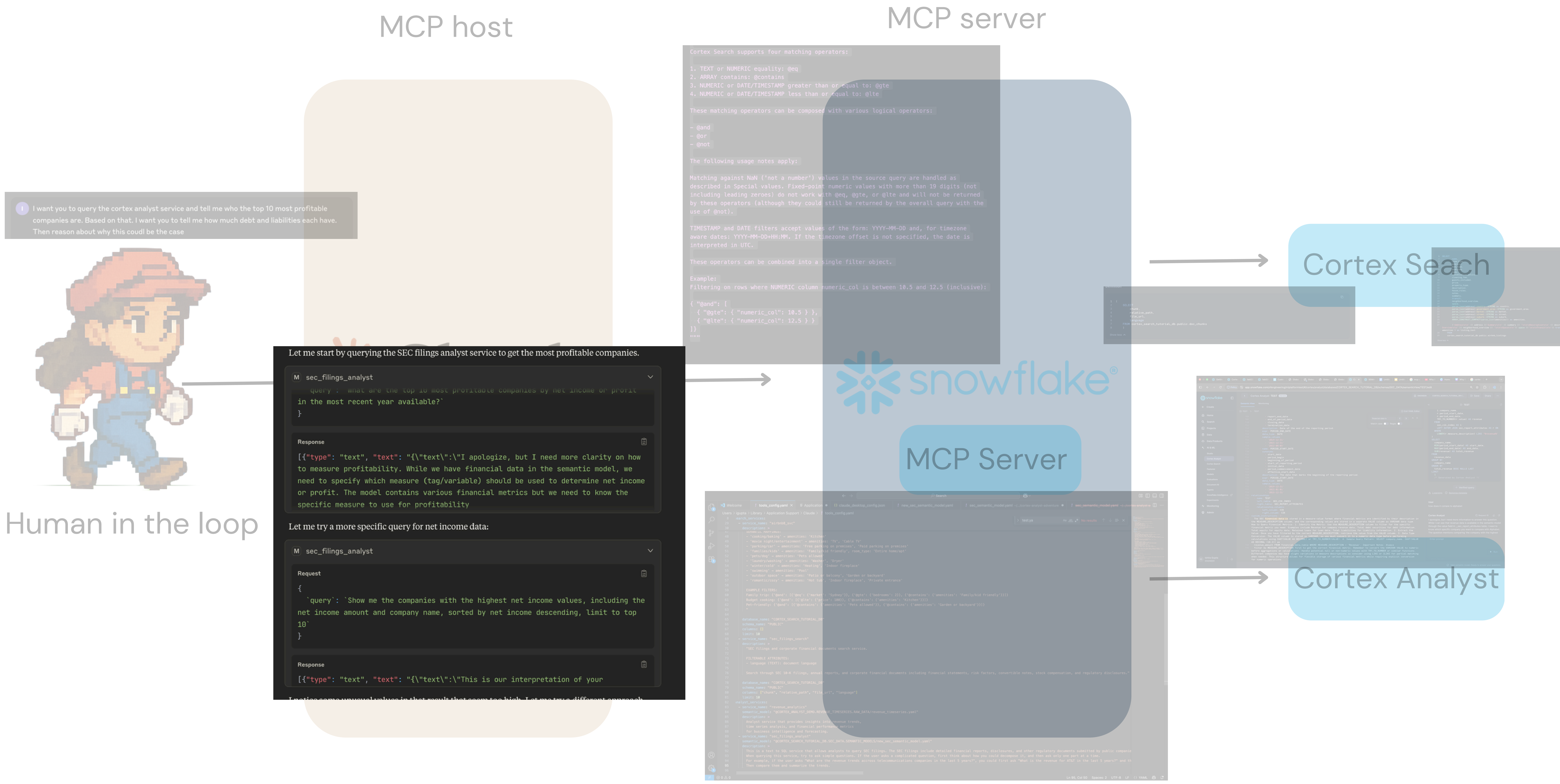
sec_filings_analyst

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

Let me start by querying the SEC filings analyst service to get the most profitable companies.

sec_filings_analyst

```
  `query`: `What are the top 10 most profitable companies by net income or profit
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

Request

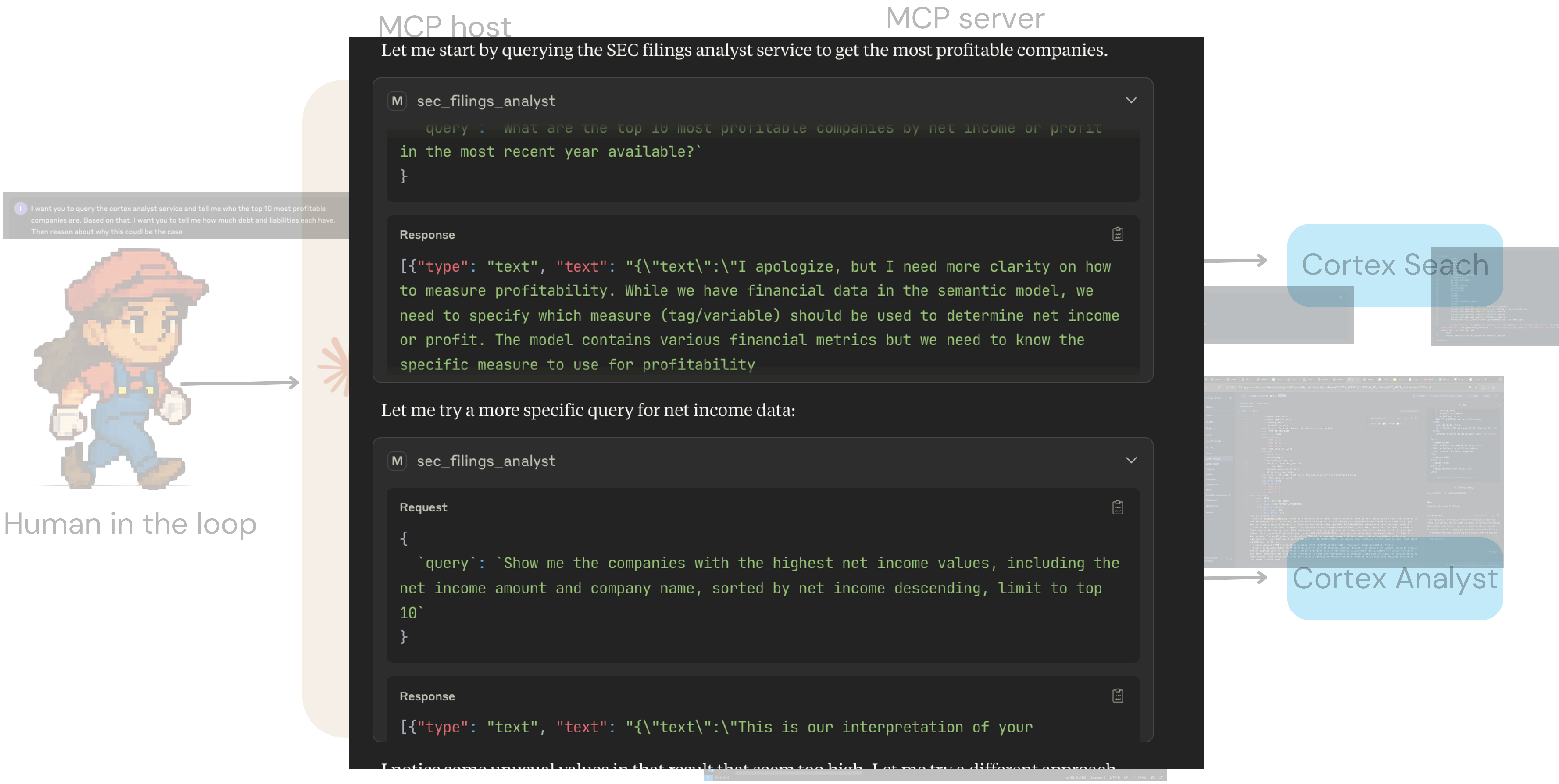
```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

I notice some unusual values in that result that seem too high. Let me try a different approach

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
  `query`: `What are the top 10 most profitable companies by net income or profit
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

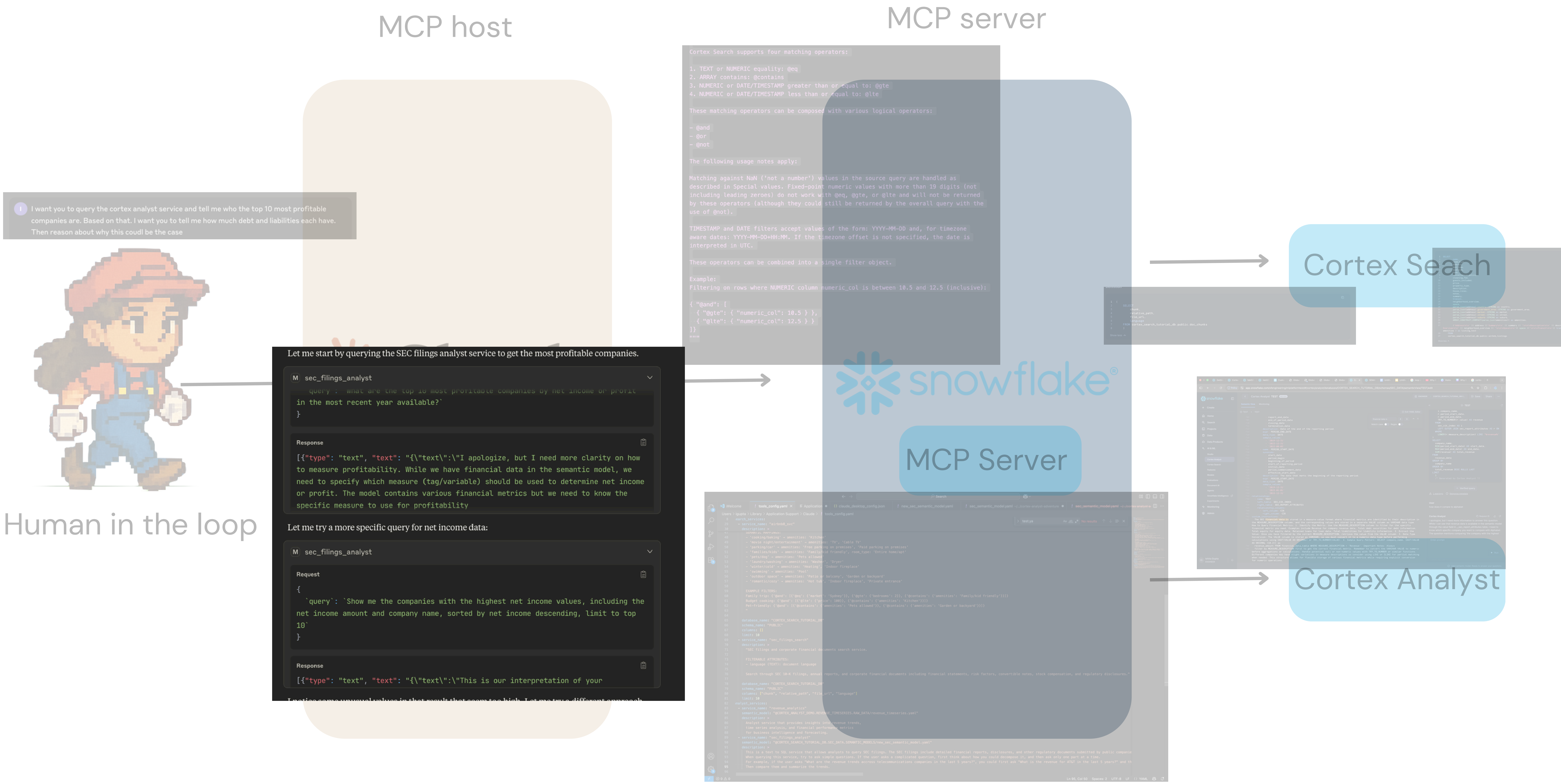
Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

I notice some unusual values in that result that seem too high. Let me try a different approach



MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

Human in the loop

Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
  `query`: `What are the top 10 most profitable companies by net income or profit
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

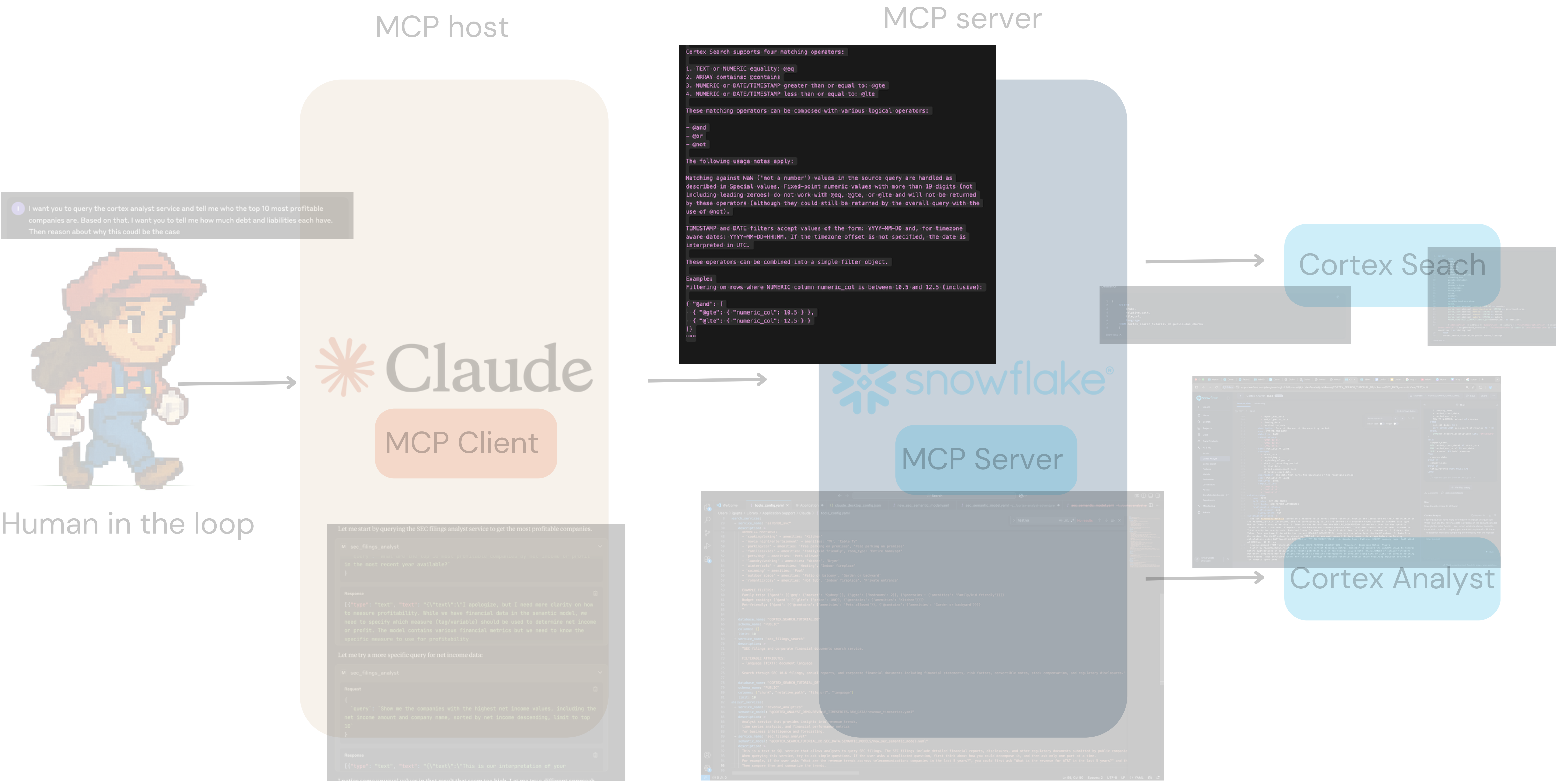
I notice some unusual values in that result that seem too high. Let me try a different approach

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

MCP Server

Cortex Seach

Cortex Analyst



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

Let me start by querying the SEC filings analyst service to get the most profitable companies.

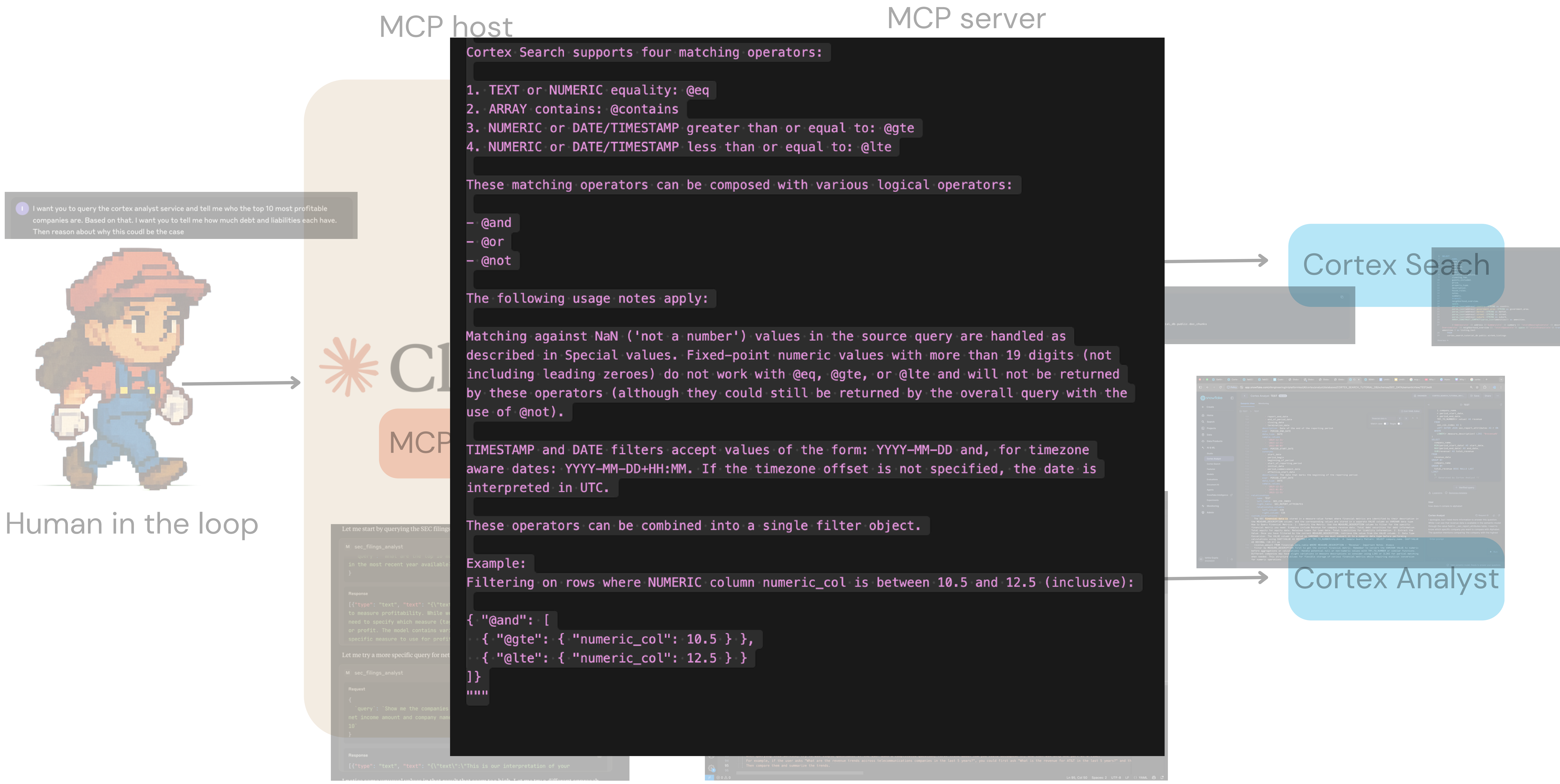
```
in the most recent year available?`
}
```

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

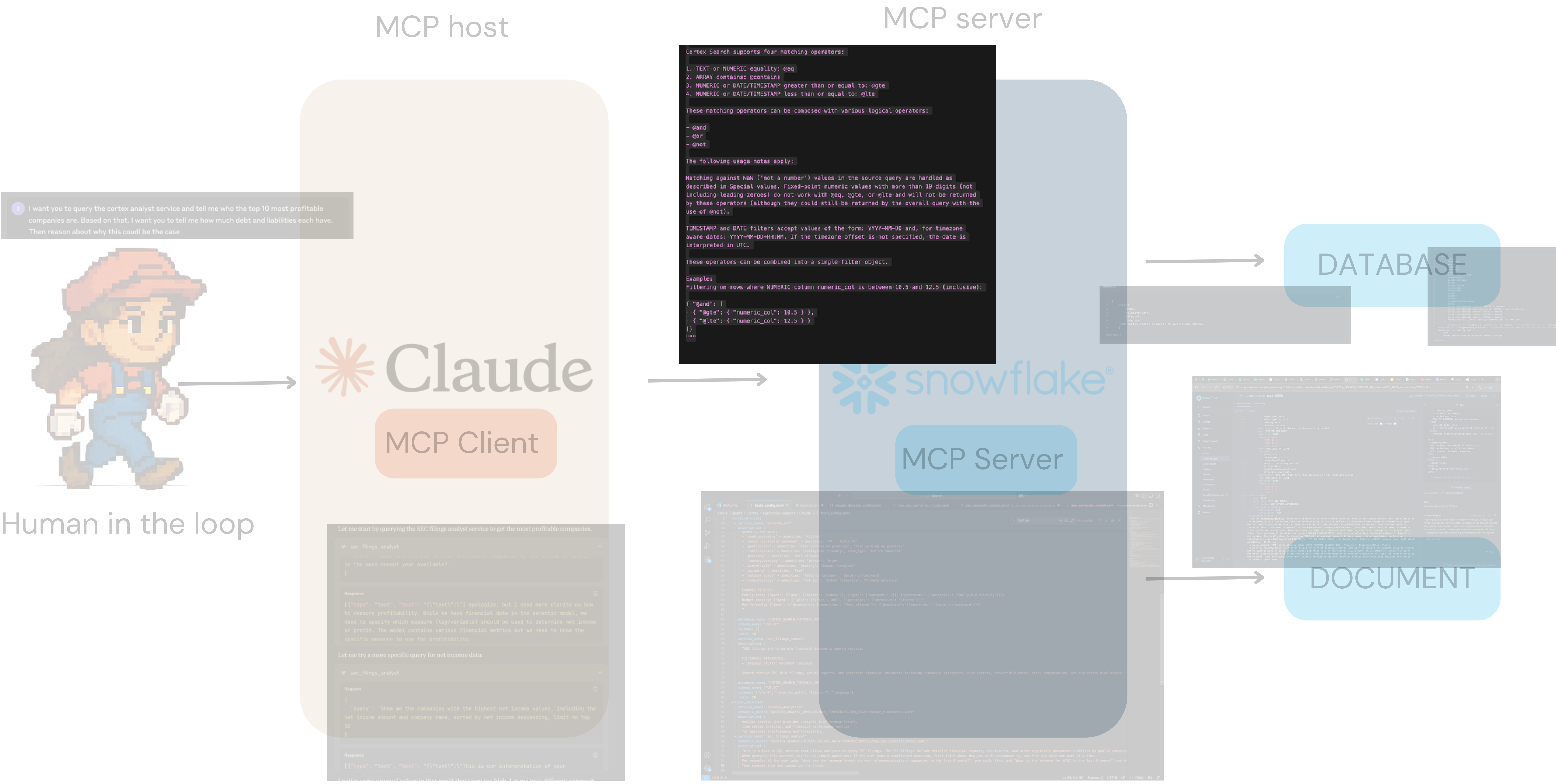
```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

Let me start by querying the SEC filings analyst service to get the most profitable companies.

sec_filings_analyst

```
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

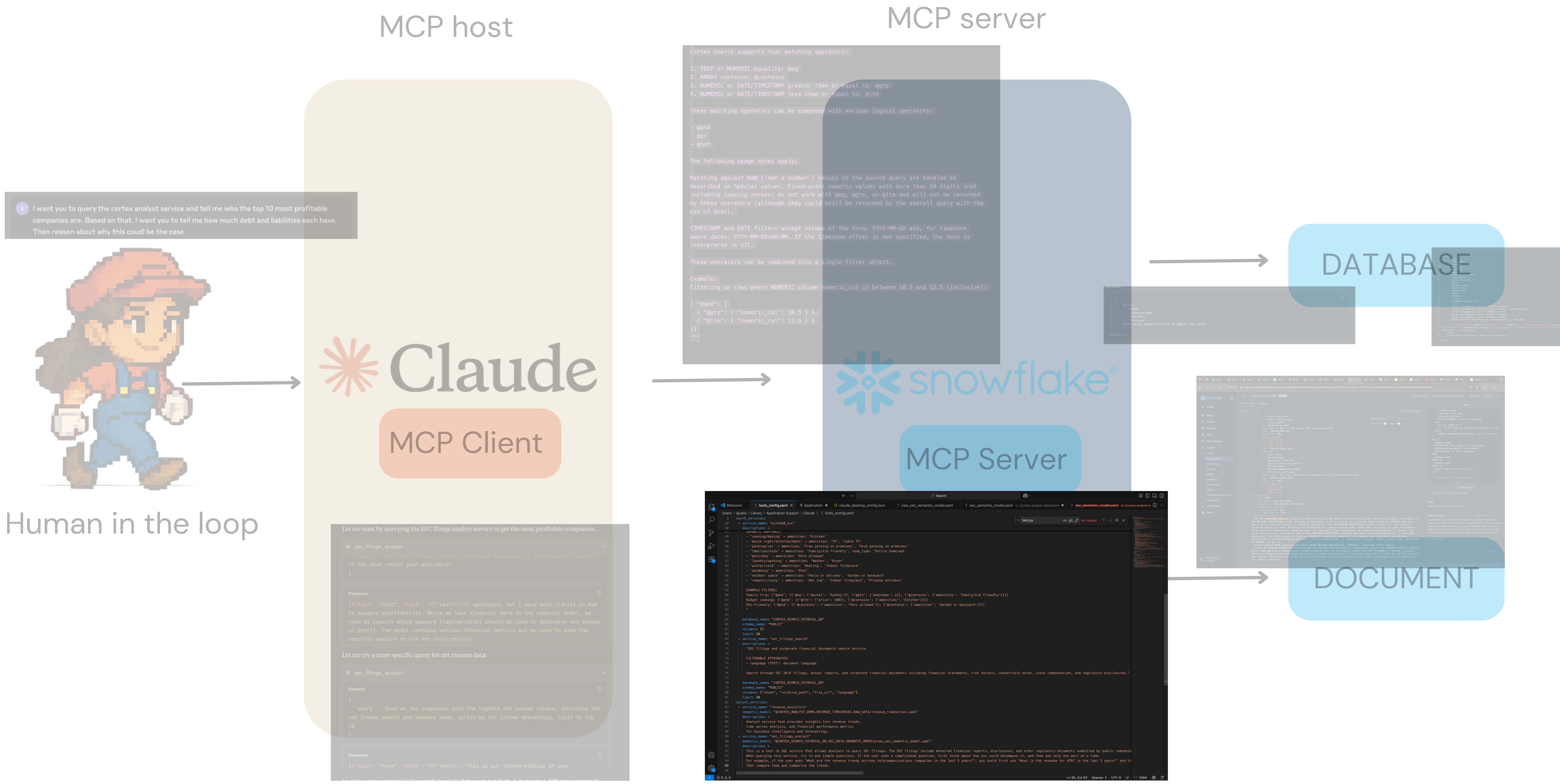
sec_filings_analyst

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

Let me start by querying the SEC filings analyst service to get the most profitable companies.

sec_filings_analyst

```
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10`
}
```

Response

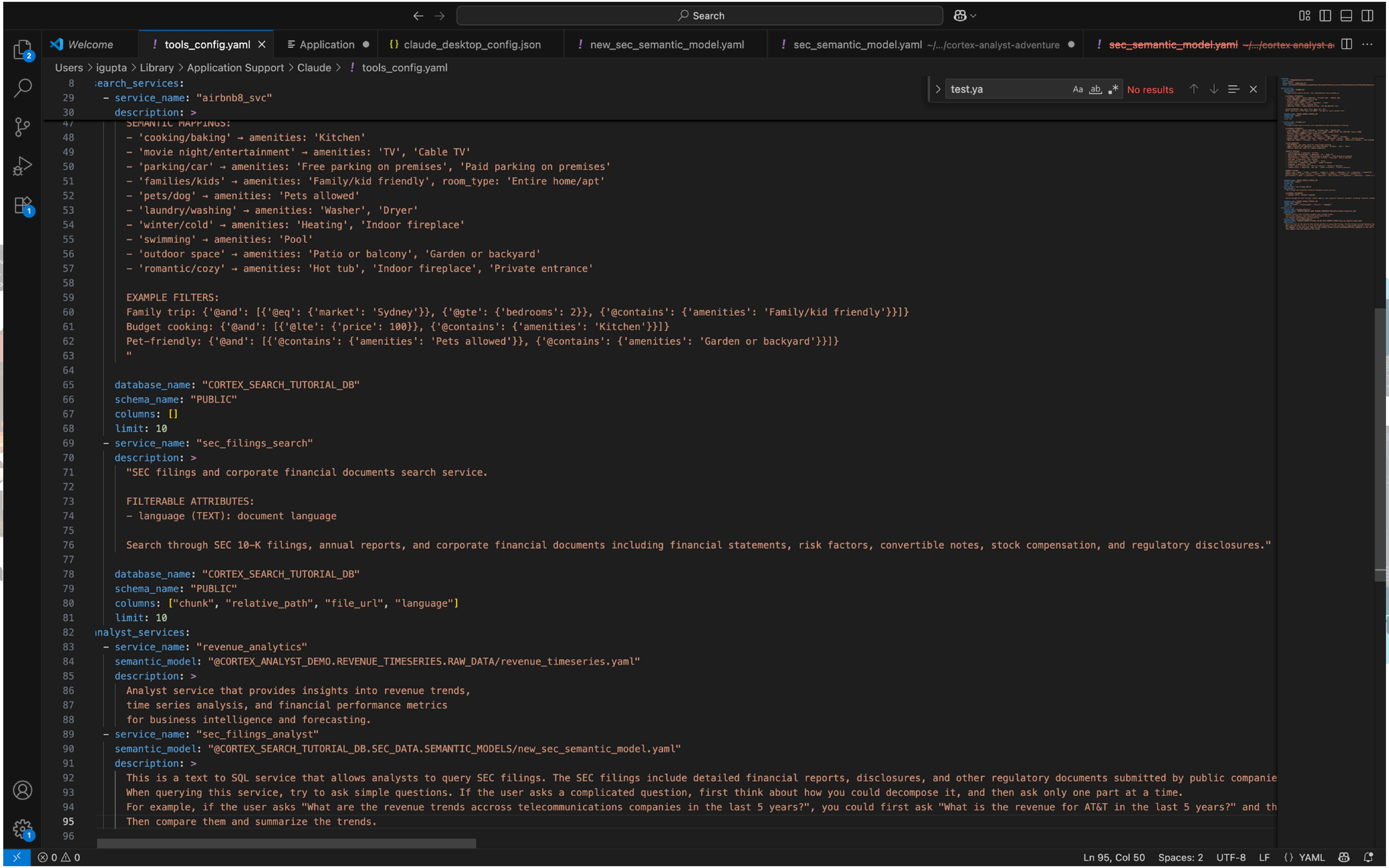
```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

Welcome | tools_config.yaml | Application | claude_desktop_config.json | new_sec_semantic_model.yaml | sec_semantic_model.yaml ~/.../cortex-analyst-adventure | ~~sec_semantic_model.yaml ~/.../cortex-analyst-a~~

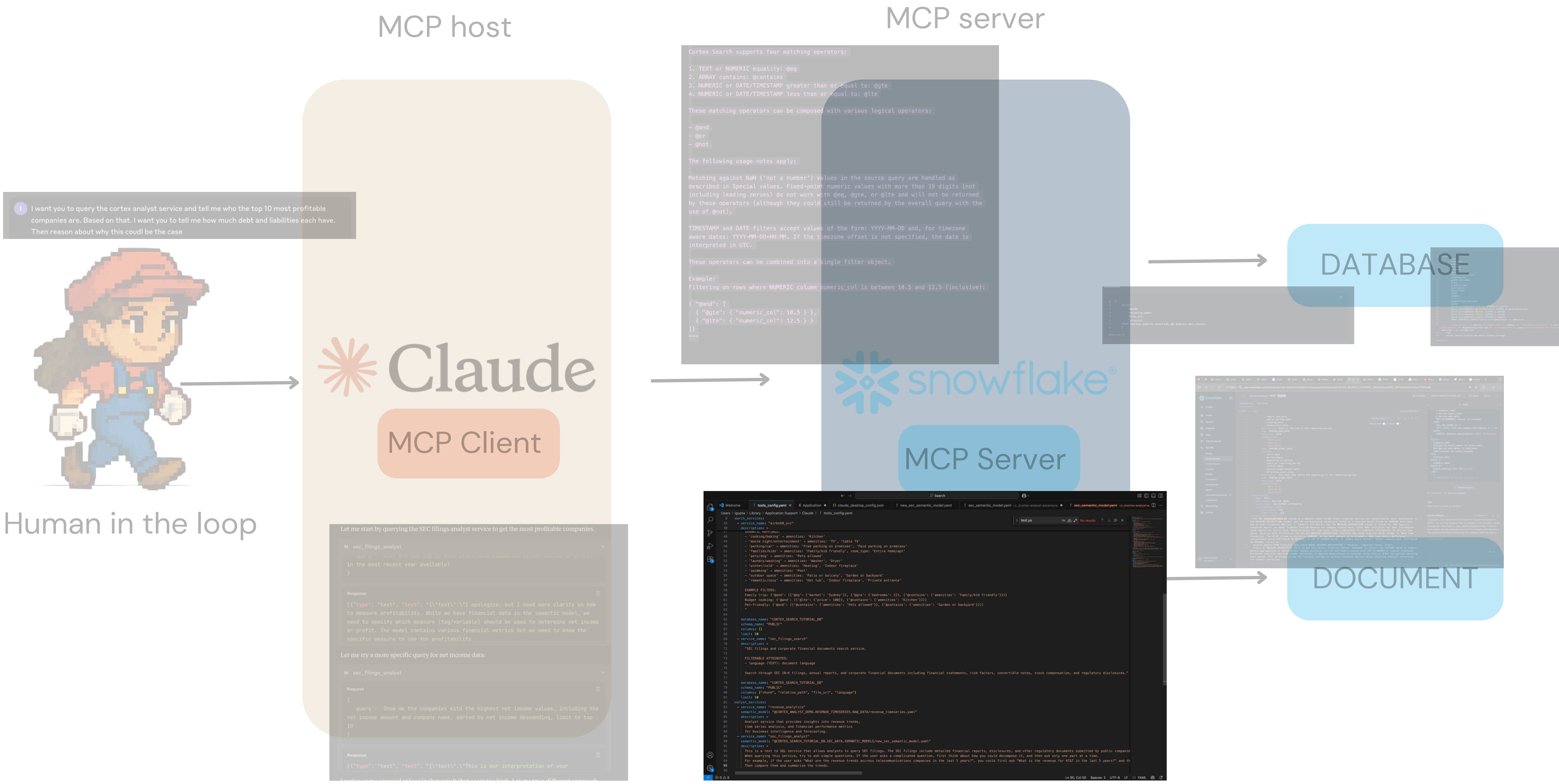
Users > igupta > Library > Application Support > Claude > tools_config.yaml

```
earch_services:
  - service_name: "airbnb8_svc"
    description: >
      SEMANTIC MAPPINGS:
      - 'cooking/baking' → amenities: 'Kitchen'
      - 'movie night/entertainment' → amenities: 'TV', 'Cable TV'
      - 'parking/car' → amenities: 'Free parking on premises', 'Paid parking on premises'
      - 'families/kids' → amenities: 'Family/kid friendly', room_type: 'Entire home/apt'
      - 'pets/dog' → amenities: 'Pets allowed'
      - 'laundry/washing' → amenities: 'Washer', 'Dryer'
      - 'winter/cold' → amenities: 'Heating', 'Indoor fireplace'
      - 'swimming' → amenities: 'Pool'
      - 'outdoor space' → amenities: 'Patio or balcony', 'Garden or backyard'
      - 'romantic/cozy' → amenities: 'Hot tub', 'Indoor fireplace', 'Private entrance'

      EXAMPLE FILTERS:
      Family trip: {'@and': [{'@eq': {'market': 'Sydney'}}, {'@gte': {'bedrooms': 2}}, {'@contains': {'amenities': 'Family/kid friendly'}}]}
      Budget cooking: {'@and': [{'@lte': {'price': 100}}, {'@contains': {'amenities': 'Kitchen'}}]}
      Pet-friendly: {'@and': [{'@contains': {'amenities': 'Pets allowed'}}, {'@contains': {'amenities': 'Garden or backyard'}}]}
      "

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: []
    limit: 10
  - service_name: "sec_filings_search"
    description: >
      "SEC filings and corporate financial documents search service.

      FILTERABLE ATTRIBUTES:
      - language (TEXT): document language

      Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: ["chunk", "relative_path", "file_url", "language"]
    limit: 10
nalyst_services:
  - service_name: "revenue_analytics"
    semantic_model: "@CORTEX_ANALYST_DEMO.REVENUE_TIMESERIES.RAW_DATA/revenue_timeseries.yaml"
    description: >
      Analyst service that provides insights into revenue trends,
      time series analysis, and financial performance metrics
      for business intelligence and forecasting.
  - service_name: "sec_filings_analyst"
    semantic_model: "@CORTEX_SEARCH_TUTORIAL_DB.SEC_DATA.SEMANTIC_MODELS/new_sec_semantic_model.yaml"
    description: >
      This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companie
      When querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time.
      For example, if the user asks "What are the revenue trends accross telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and th
      Then compare them and summarize the trends.
```

Ln 95, Col 50   Spaces: 2   UTF-8   LF   YAML


Search
Welcome
tools_config.yaml
Application
claude_desktop_config.json
new_sec_semantic_model.yaml
sec_semantic_model.yaml ~/.../cortex-analyst-adventure
~~sec_semantic_model.yaml ~/.../cortex-analyst-a~~
Users > igupta > Library > Application Support > Claude > tools_config.yaml
test.ya
Aa
No results
Ln 95, Col 50
Spaces: 2
UTF-8
LF
YAML

```
earch_services:
  - service_name: "airbnb8_svc"
    description: >
      SEMANTIC MAPPINGS:
      - 'cooking/baking' → amenities: 'Kitchen'
      - 'movie night/entertainment' → amenities: 'TV', 'Cable TV'
      - 'parking/car' → amenities: 'Free parking on premises', 'Paid parking on premises'
      - 'families/kids' → amenities: 'Family/kid friendly', room_type: 'Entire home/apt'
      - 'pets/dog' → amenities: 'Pets allowed'
      - 'laundry/washing' → amenities: 'Washer', 'Dryer'
      - 'winter/cold' → amenities: 'Heating', 'Indoor fireplace'
      - 'swimming' → amenities: 'Pool'
      - 'outdoor space' → amenities: 'Patio or balcony', 'Garden or backyard'
      - 'romantic/cozy' → amenities: 'Hot tub', 'Indoor fireplace', 'Private entrance'

      EXAMPLE FILTERS:
      Family trip: {'@and': [{'@eq': {'market': 'Sydney'}}, {'@gte': {'bedrooms': 2}}, {'@contains': {'amenities': 'Family/kid friendly'}}]}
      Budget cooking: {'@and': [{'@lte': {'price': 100}}, {'@contains': {'amenities': 'Kitchen'}}]}
      Pet-friendly: {'@and': [{'@contains': {'amenities': 'Pets allowed'}}, {'@contains': {'amenities': 'Garden or backyard'}}]}
      "

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: []
    limit: 10
  - service_name: "sec_filings_search"
    description: >
      "SEC filings and corporate financial documents search service.

      FILTERABLE ATTRIBUTES:
      - language (TEXT): document language

      Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: ["chunk", "relative_path", "file_url", "language"]
    limit: 10
nalyst_services:
  - service_name: "revenue_analytics"
    semantic_model: "@CORTEX_ANALYST_DEMO.REVENUE_TIMESERIES.RAW_DATA/revenue_timeseries.yaml"
    description: >
      Analyst service that provides insights into revenue trends,
      time series analysis, and financial performance metrics
      for business intelligence and forecasting.
  - service_name: "sec_filings_analyst"
    semantic_model: "@CORTEX_SEARCH_TUTORIAL_DB.SEC_DATA.SEMANTIC_MODELS/new_sec_semantic_model.yaml"
    description: >
      This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companie
      When querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time.
      For example, if the user asks "What are the revenue trends accross telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and th
      Then compare them and summarize the trends.

```



I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

Let me start by querying the SEC filings analyst service to get the most profitable companies.

sec_filings_analyst

```
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10`
}
```

Response

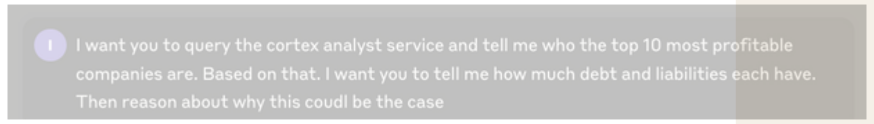
```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

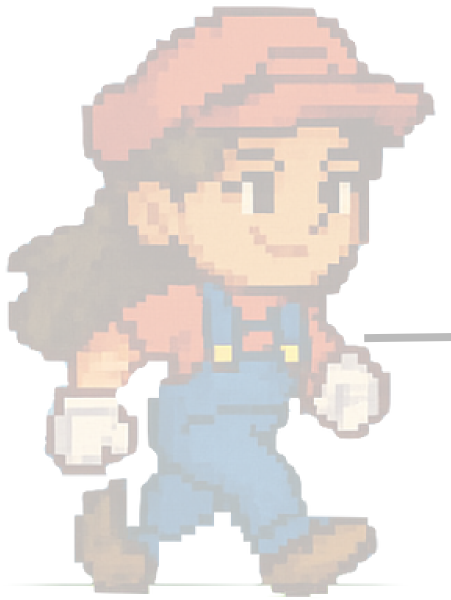
Users > igupta > Library > Application Support > Claude > tools_config.yaml

```
search_services:
  - service_name: "airbnb8_svc"
    description: >
      SEMANTIC MAPPINGS:
      - 'cooking/baking' → amenities: 'Kitchen'
      - 'movie night/entertainment' → amenities: 'TV', 'Cable TV'
      - 'parking/car' → amenities: 'Free parking on premises', 'Paid parking on premises'
      - 'families/kids' → amenities: 'Family/kid friendly', room_type: 'Entire home/apt'
      - 'pets/dog' → amenities: 'Pets allowed'
      - 'laundry/washing' → amenities: 'Washer', 'Dryer'
      - 'winter/cold' → amenities: 'Heating', 'Indoor fireplace'
      - 'swimming' → amenities: 'Pool'
      - 'outdoor space' → amenities: 'Patio or balcony', 'Garden or backyard'
      - 'romantic/cozy' → amenities: 'Hot tub', 'Indoor fireplace', 'Private entrance'

      EXAMPLE FILTERS:
      Family trip: {'@and': [{'@eq': {'market': 'Sydney'}}, {'@gte': {'bedrooms': 2}}, {'@contains': {'amenities': 'Family/kid friendly'}}]}
      Budget cooking: {'@and': [{'@lte': {'price': 100}}, {'@contains': {'amenities': 'Kitchen'}}]}
      Pet-friendly: {'@and': [{'@contains': {'amenities': 'Pets allowed'}}, {'@contains': {'amenities': 'Garden or backyard'}}]}
      "

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: []
    limit: 10
  - service_name: "sec_filings_search"
    description: >
      "SEC filings and corporate financial documents search service.

      FILTERABLE ATTRIBUTES:
      - language (TEXT): document language

      Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: ["chunk", "relative_path", "file_url", "language"]
    limit: 10
analyst_services:
  - service_name: "revenue_analytics"
    semantic_model: "@CORTEX_ANALYST_DEMO.REVENUE_TIMESERIES.RAW_DATA/revenue_timeseries.yaml"
    description: >
      Analyst service that provides insights into revenue trends,
      time series analysis, and financial performance metrics
      for business intelligence and forecasting.
  - service_name: "sec_filings_analyst"
    semantic_model: "@CORTEX_SEARCH_TUTORIAL_DB.SEC_DATA.SEMANTIC_MODELS/new_sec_semantic_model.yaml"
    description: >
      This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companie
      When querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time.
      For example, if the user asks "What are the revenue trends accross telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and th
      Then compare them and summarize the trends.
```

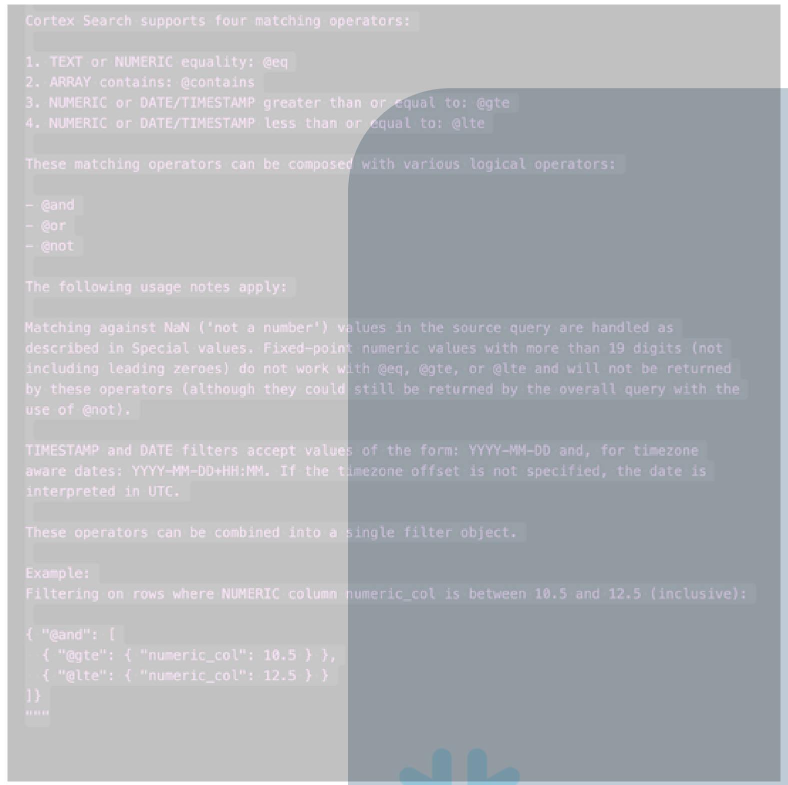
MCP host

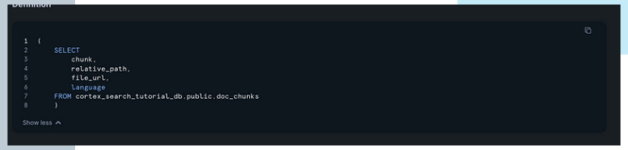
MCP server

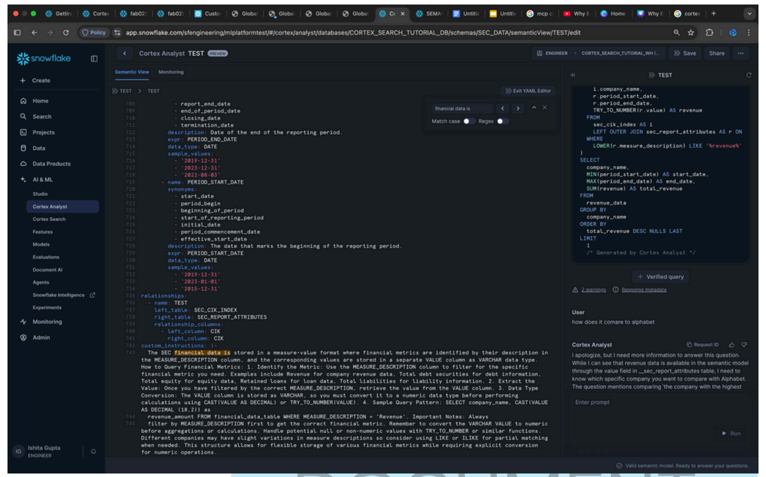

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that. I want you to tell me how much debt and liabilities each have. Then reason about why this coudl be the case

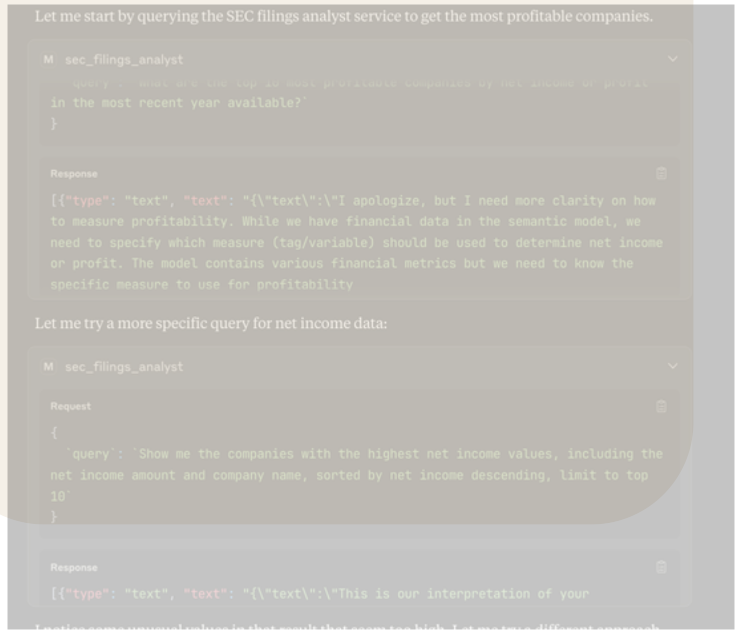


Human in the loop


```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
by these operators (although they could still be returned by the overall query with the
use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone
aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is
interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```


```
(
    SELECT
        chunk,
        relative_path,
        file_url,
        language
    FROM cortex_search_tutorial_db.public.doc_chunks
    )
```




Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability
```

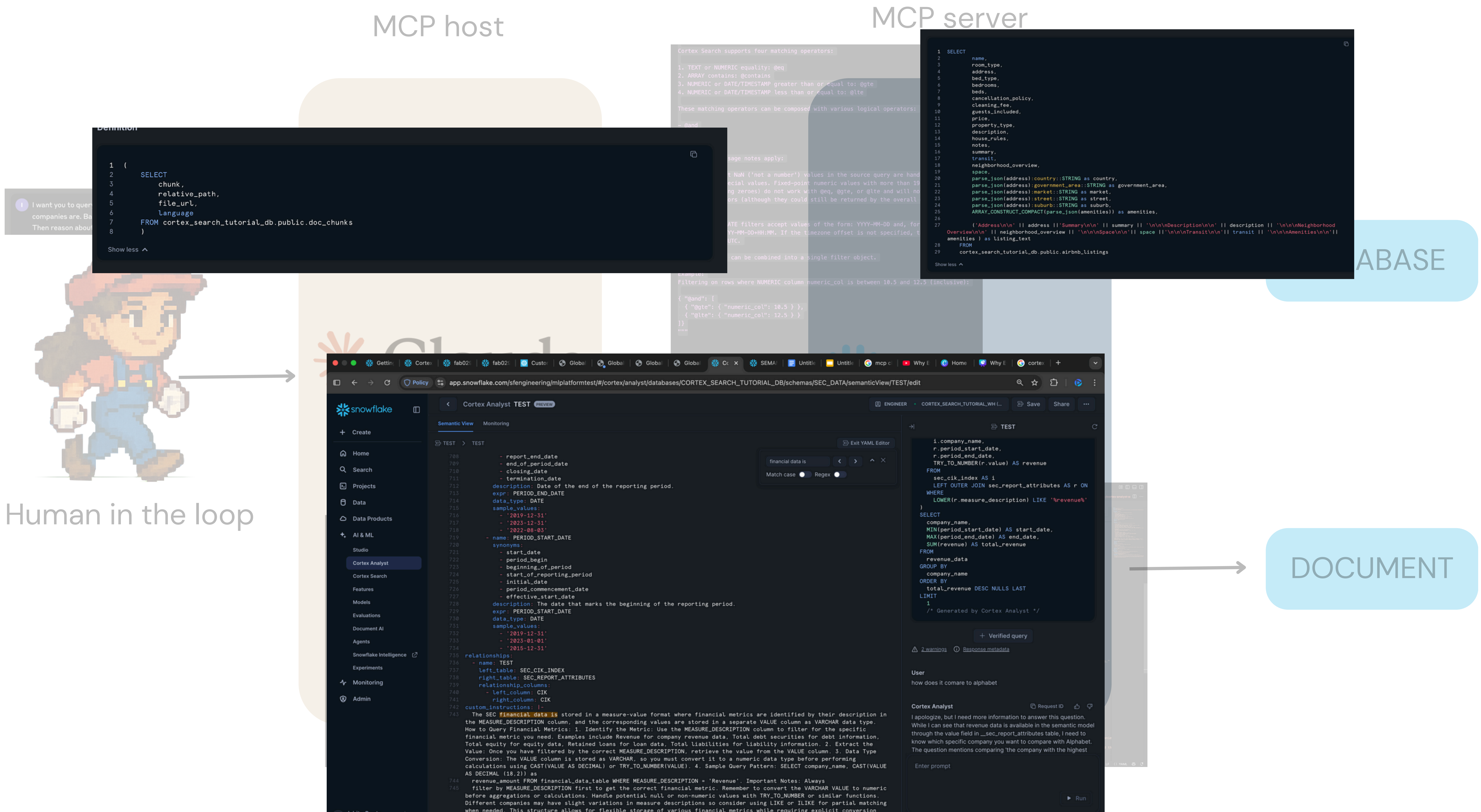
Let me try a more specific query for net income data:

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```



MCP host

MCP server

Human in the loop

ABASE

DOCUMENT

I want you to quer
companies are. Ba
Then reason about

Definition

```
(
    SELECT
        chunk,
        relative_path,
        file_url,
        language
    FROM cortex_search_tutorial_db.public.doc_chunks
    )
```

Show less

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and

sage notes apply:

t NaN ('not a number') values in the source query are hand
ecial values. Fixed-point numeric values with more than 19
ng zeroes) do not work with @eq, @gte, or @lte and will no
ors (although they could still be returned by the overall

ATE filters accept values of the form: YYYY-MM-DD and, for
YY-MM-DD+HH:MM. If the timezone offset is not specified, t
UTC.

 can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):

{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
"""
```

```
SELECT
        name,
        room_type,
        address,
        bed_type,
        bedrooms,
        beds,
        cancellation_policy,
        cleaning_fee,
        guests_included,
        price,
        property_type,
        description,
        house_rules,
        notes,
        summary,
        transit,
        neighborhood_overview,
        space,
        parse_json(address):country::STRING as country,
        parse_json(address):government_area::STRING as government_area,
        parse_json(address):market::STRING as market,
        parse_json(address):street::STRING as street,
        parse_json(address):suburb::STRING as suburb,
        ARRAY_CONSTRUCT_COMPACT(parse_json(amenities)) as amenities,

        ('Address\n\n' || address ||'Summary\n\n' || summary || '\n\n\nDescription\n\n' || description || '\n\n\nNeighborhood
Overview\n\n' || neighborhood_overview || '\n\n\nSpace\n\n'|| space ||'\n\n\nTransit\n\n'|| transit || '\n\n\nAmenities\n\n'||
amenities ) as listing_text
    FROM
    cortex_search_tutorial_db.public.airbnb_listings
```

Show less

app.snowflake.com/sfengineering/mlplatformtest/#/cortex/analyst/databases/CORTEX_SEARCH_TUTORIAL_DB/schemas/SEC_DATA/semanticView/TEST/edit

```
                - report_end_date
                - end_of_period_date
                - closing_date
                - termination_date
              description: Date of the end of the reporting period.
              expr: PERIOD_END_DATE
              data_type: DATE
              sample_values:
                - '2019-12-31'
                - '2023-12-31'
                - '2022-08-03'
            - name: PERIOD_START_DATE
              synonyms:
                - start_date
                - period_begin
                - beginning_of_period
                - start_of_reporting_period
                - initial_date
                - period_commencement_date
                - effective_start_date
              description: The date that marks the beginning of the reporting period.
              expr: PERIOD_START_DATE
              data_type: DATE
              sample_values:
                - '2019-12-31'
                - '2023-01-01'
                - '2015-12-31'
relationships:
  - name: TEST
    left_table: SEC_CIK_INDEX
    right_table: SEC_REPORT_ATTRIBUTES
    relationship_columns:
      - left_column: CIK
        right_column: CIK
custom_instructions: |-
  The SEC financial data is stored in a measure-value format where financial metrics are identified by their description in
the MEASURE_DESCRIPTION column, and the corresponding values are stored in a separate VALUE column as VARCHAR data type.
How to Query Financial Metrics: 1. Identify the Metric: Use the MEASURE_DESCRIPTION column to filter for the specific
financial metric you need. Examples include Revenue for company revenue data, Total debt securities for debt information,
Total equity for equity data, Retained loans for loan data, Total liabilities for liability information. 2. Extract the
Value: Once you have filtered by the correct MEASURE_DESCRIPTION, retrieve the value from the VALUE column. 3. Data Type
Conversion: The VALUE column is stored as VARCHAR, so you must convert it to a numeric data type before performing
calculations using CAST(VALUE AS DECIMAL) or TRY_TO_NUMBER(VALUE). 4. Sample Query Pattern: SELECT company_name, CAST(VALUE
AS DECIMAL (18,2)) as
  revenue_amount FROM financial_data_table WHERE MEASURE_DESCRIPTION = 'Revenue'. Important Notes: Always
  filter by MEASURE_DESCRIPTION first to get the correct financial metric. Remember to convert the VARCHAR VALUE to numeric
before aggregations or calculations. Handle potential null or non-numeric values with TRY_TO_NUMBER or similar functions.
Different companies may have slight variations in measure descriptions so consider using LIKE or ILIKE for partial matching
when needed. This structure allows for flexible storage of various financial metrics while requiring explicit conversion
```

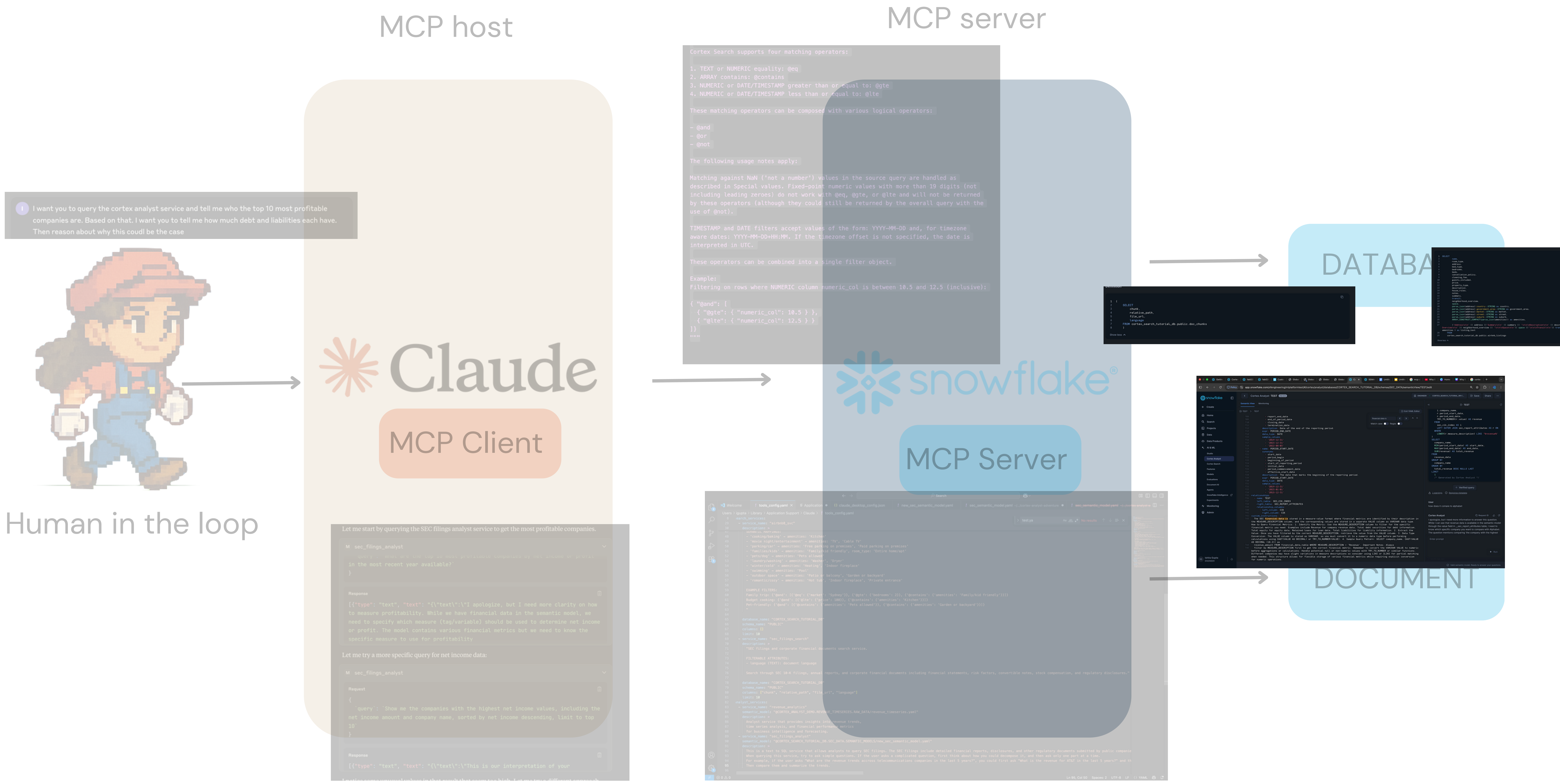
```
      i.company_name,
      r.period_start_date,
      r.period_end_date,
      TRY_TO_NUMBER(r.value) AS revenue
    FROM
      sec_cik_index AS i
      LEFT OUTER JOIN sec_report_attributes AS r ON
    WHERE
      LOWER(r.measure_description) LIKE '%revenue%'
  )
  SELECT
    company_name,
    MIN(period_start_date) AS start_date,
    MAX(period_end_date) AS end_date,
    SUM(revenue) AS total_revenue
  FROM
    revenue_data
  GROUP BY
    company_name
  ORDER BY
    total_revenue DESC NULLS LAST
  LIMIT
    1
    /* Generated by Cortex Analyst */
```

User

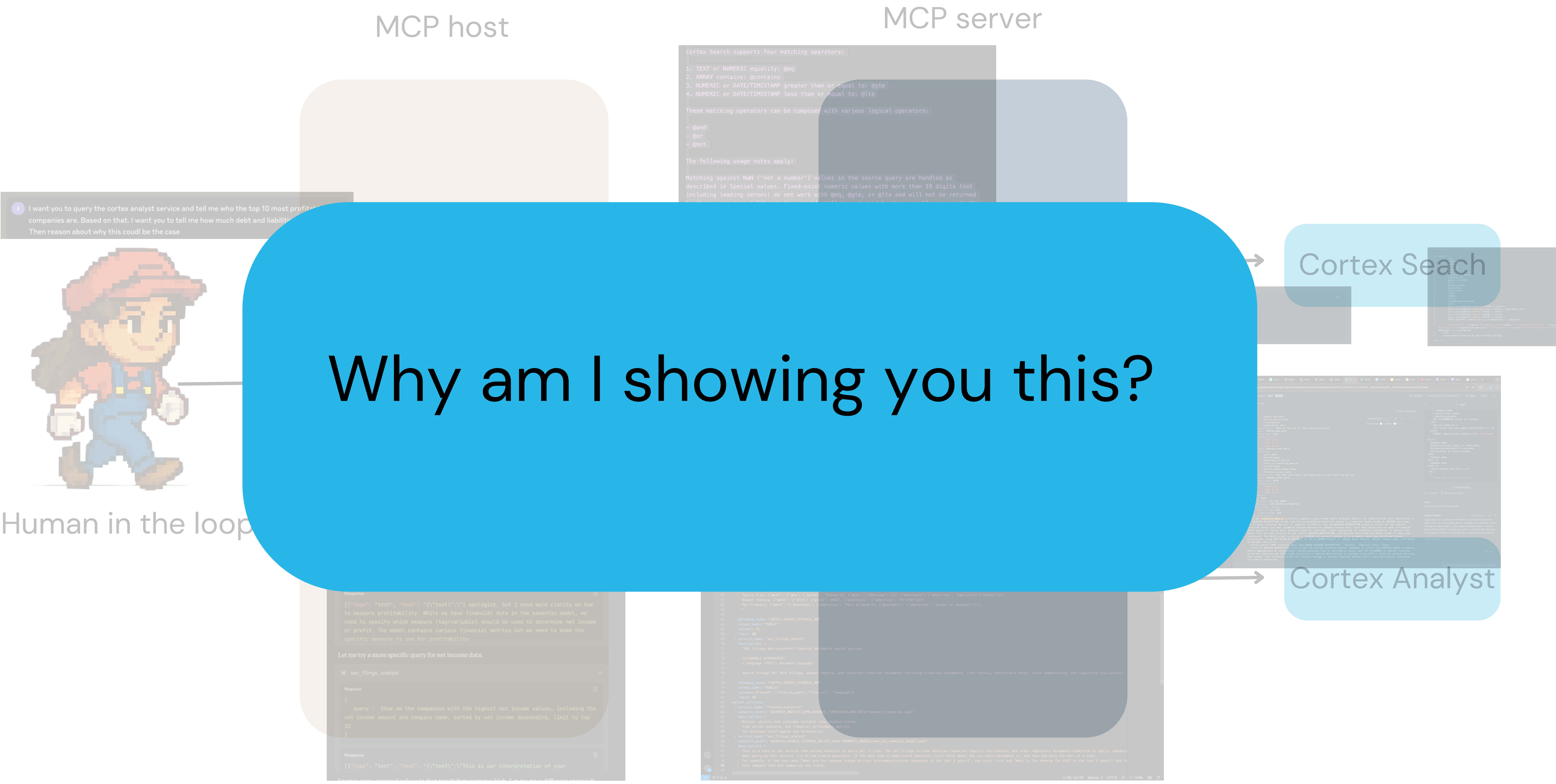
how does it comare to alphabet

Cortex Analyst

I apologize, but I need more information to answer this question. While I can see that revenue data is available in the semantic model through the value field in __sec_report_attributes table, I need to know which specific company you want to compare with Alphabet. The question mentions comparing 'the company with the highest



Why am I showing you this?



I want you to query the cortex analyst service and tell me who the top 10 most profit
companies are. Based on that, I want you to tell me how much debt and liabiliti
Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

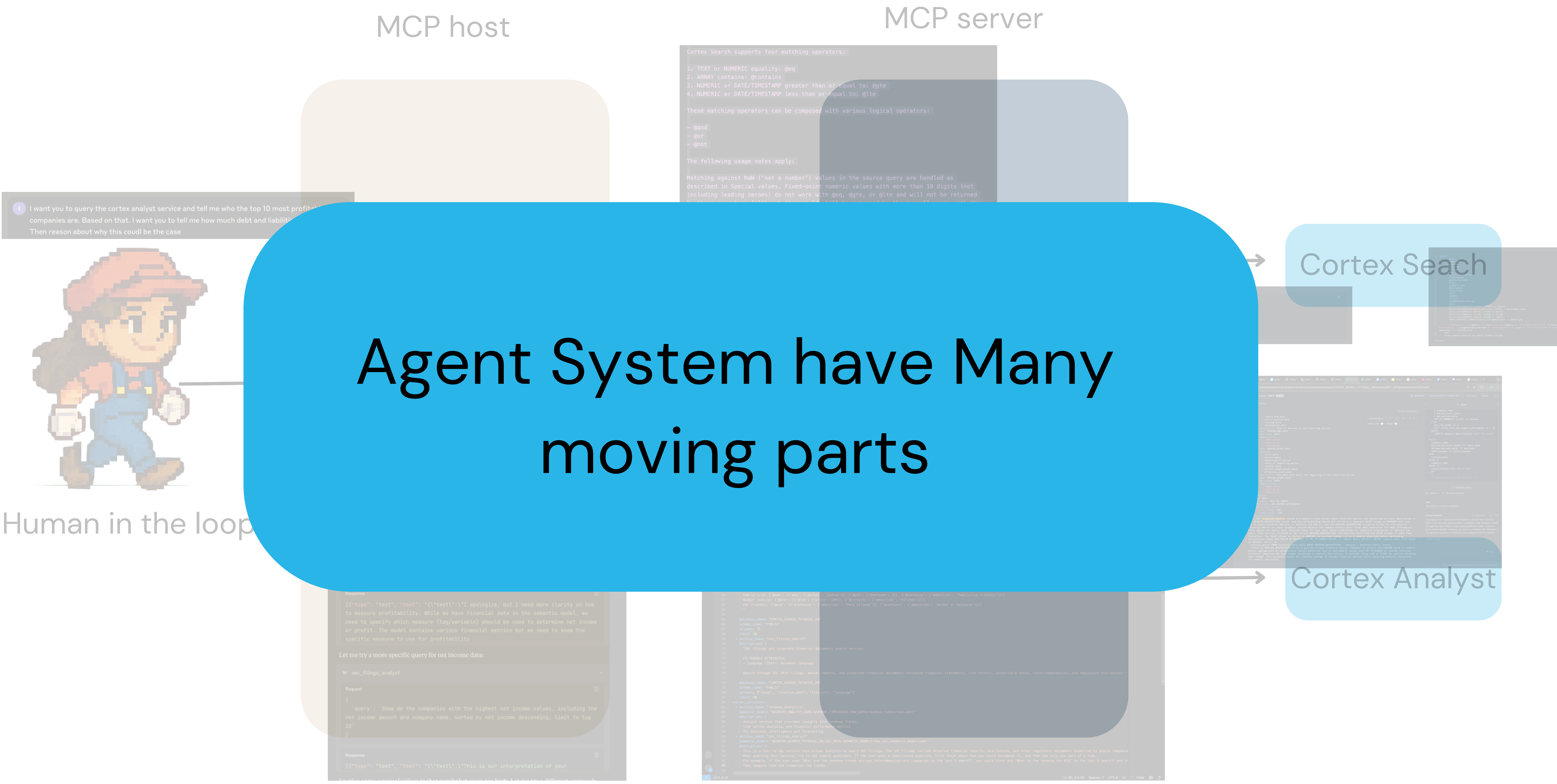
Request

```
{
  `query`: `Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

Agent System have Many moving parts



I want you to query the cortex analyst service and tell me who the top 10 most profit
companies are. Based on that. I want you to tell me how much debt and liabiliti
Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

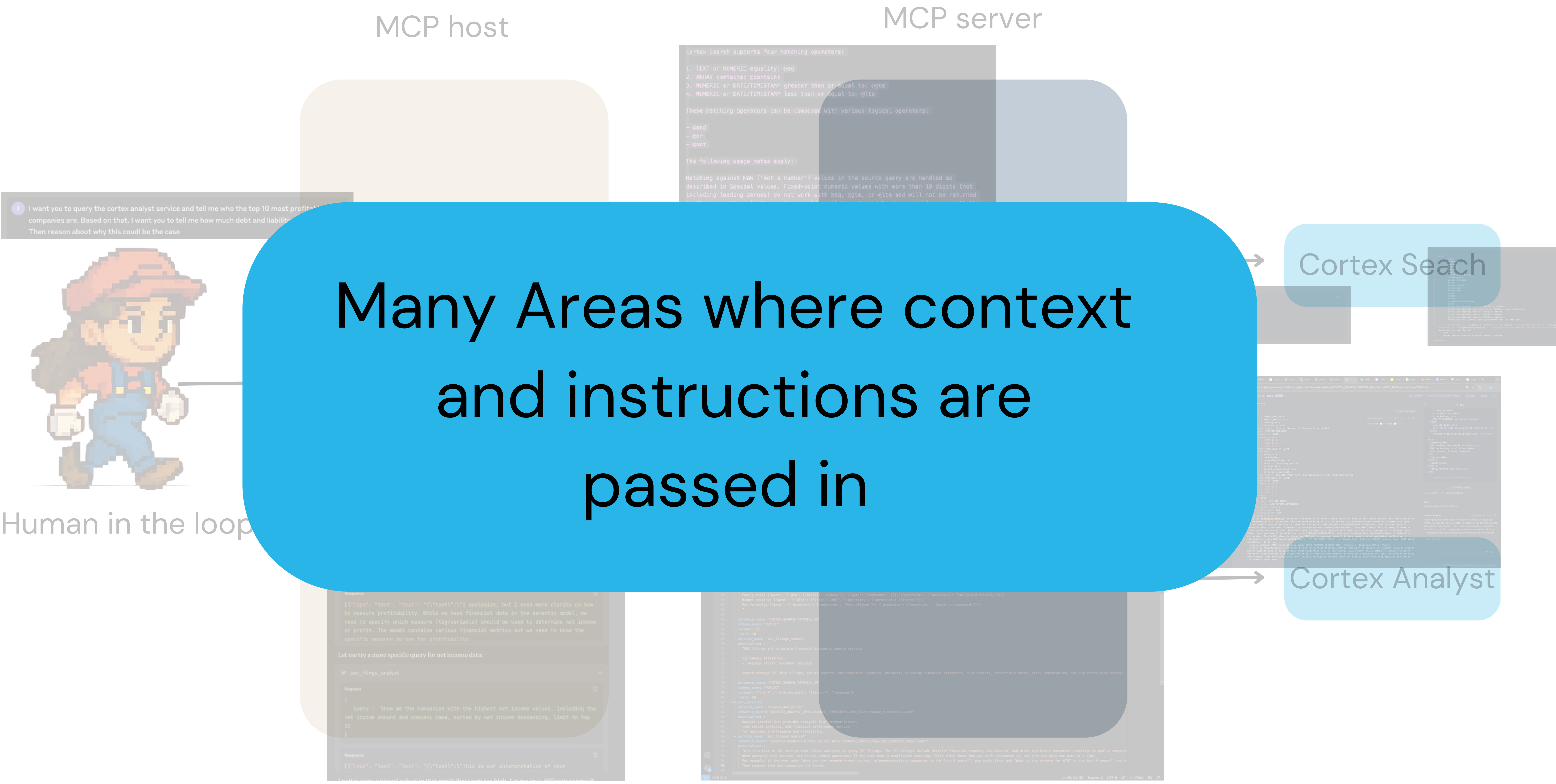
Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

Many Areas where context and instructions are passed in



I want you to query the cortex analyst service and tell me who the top 10 most profita
companies are. Based on that. I want you to tell me how much debt and liabiliti
Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

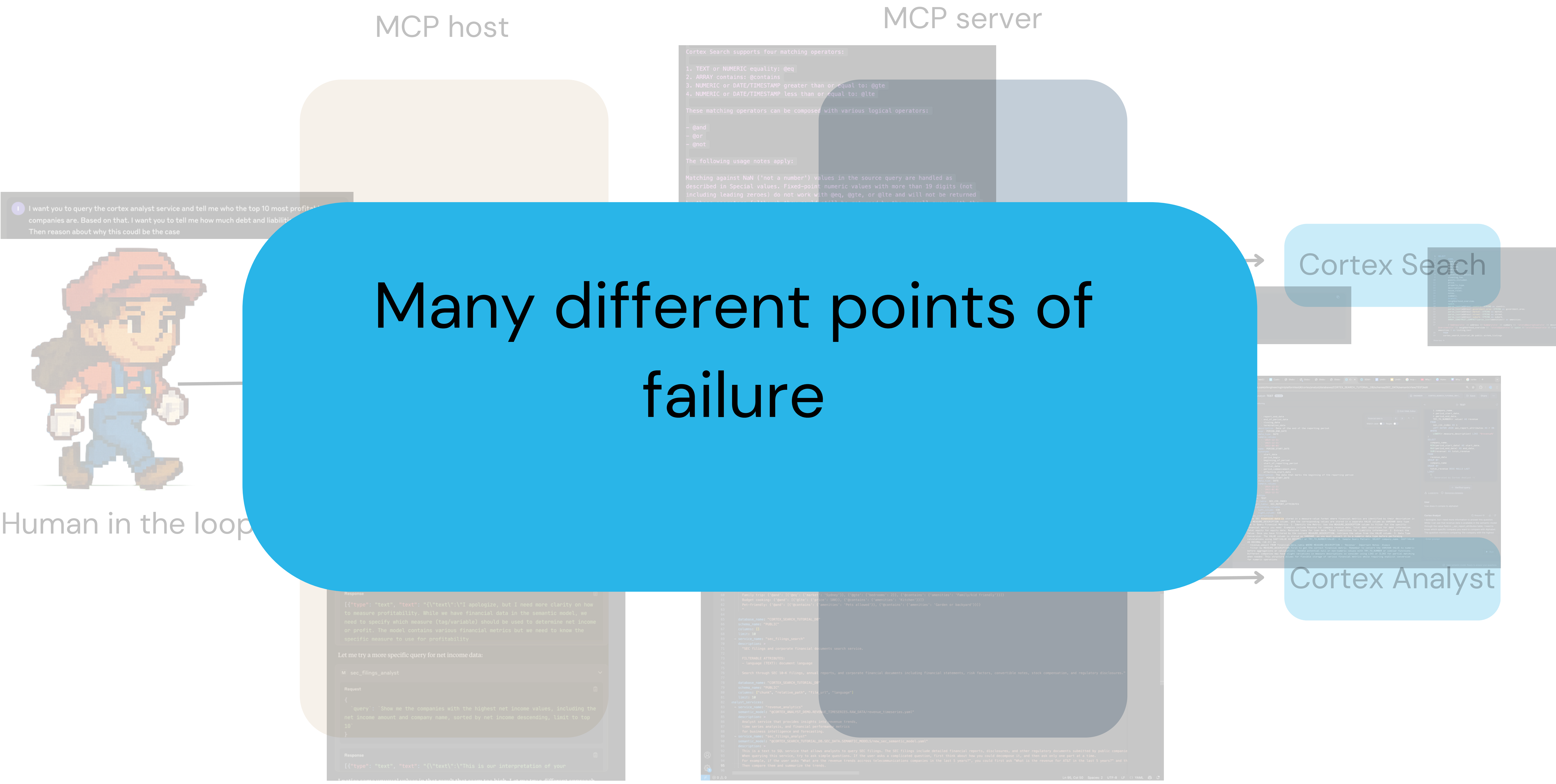
Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

```
      Family trip: {'@and': [{'@eq': {'market': 'Sydney'}}, {'@gte': {'bedrooms': 2}}, {'@contains': {'amenities': 'Family/kid friendly'}}]}
      Budget cooking: {'@and': [{'@lte': {'price': 100}}, {'@contains': {'amenities': 'Kitchen'}}]}
      Pet-friendly: {'@and': [{'@contains': {'amenities': 'Pets allowed'}}, {'@contains': {'amenities': 'Garden or backyard'}}]}
      "

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: []
    limit: 10
  - service_name: "sec_filings_search"
    description: >
      "SEC filings and corporate financial documents search service.

      FILTERABLE ATTRIBUTES:
      - language (TEXT): document language

      Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: ["chunk", "relative_path", "file_url", "language"]
    limit: 10
analyst_services:
  - service_name: "revenue_analytics"
    semantic_model: "@CORTEX_ANALYST_DEMO.REVENUE_TIMESERIES.RAW_DATA/revenue_timeseries.yaml"
    description: >
      Analyst service that provides insights into revenue trends,
      time series analysis, and financial performance metrics
      for business intelligence and forecasting.
  - service_name: "sec_filings_analyst"
    semantic_model: "@CORTEX_SEARCH_TUTORIAL_DB.SEC_DATA.SEMANTIC_MODELS/new_sec_semantic_model.yaml"
    description: >
      This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companie
      When querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time.
      For example, if the user asks "What are the revenue trends accross telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and th
      Then compare them and summarize the trends.
```



I want you to query the cortex analyst service and tell me who the top 10 most profit

companies are. Based on that. I want you to tell me how much debt and liabiliti

Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

Request

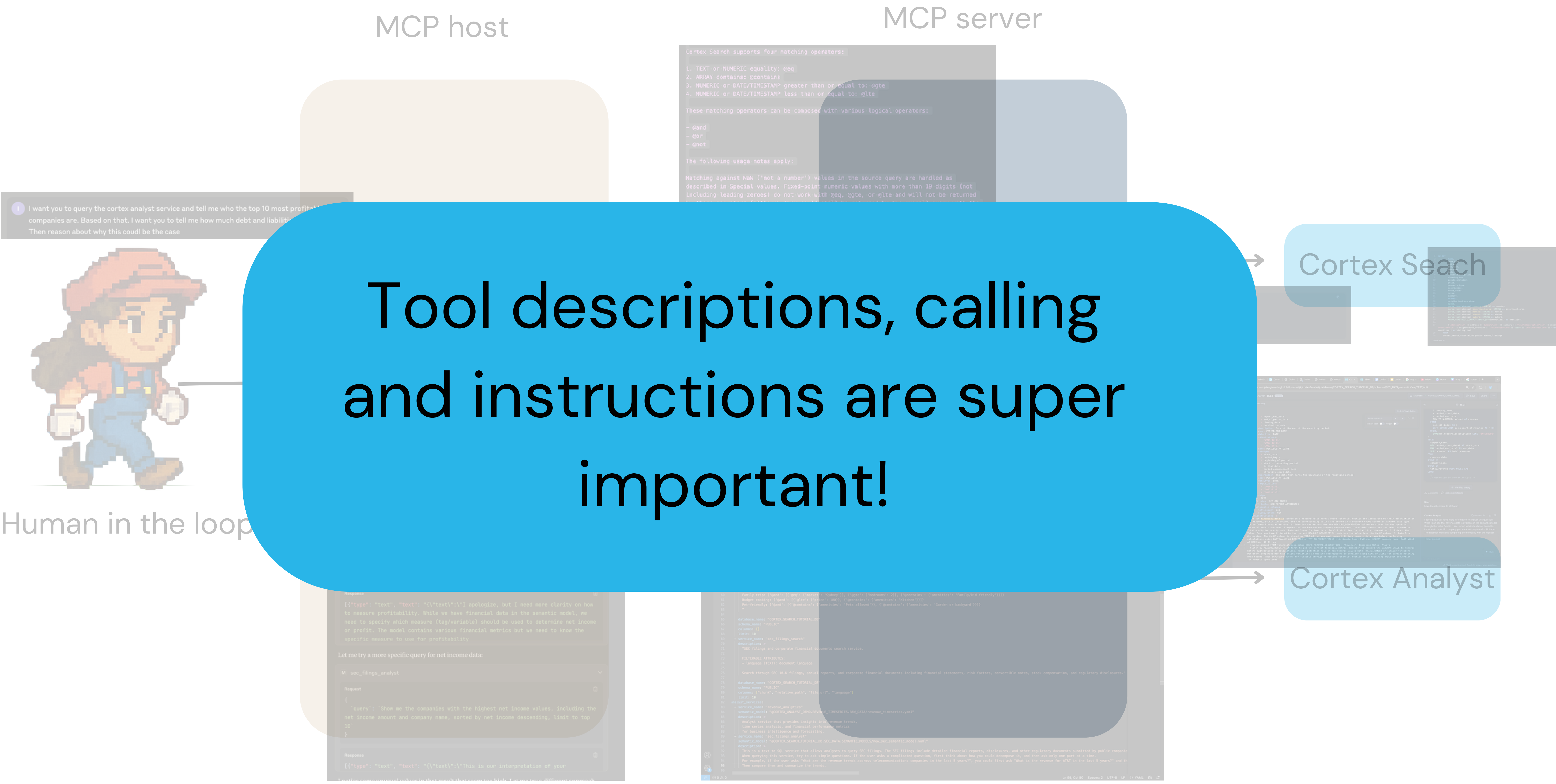
```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

Many different points of failure

Tool descriptions, calling and instructions are super important!



I want you to query the cortex analyst service and tell me who the top 10 most profit
companies are. Based on that. I want you to tell me how much debt and liabiliti
Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

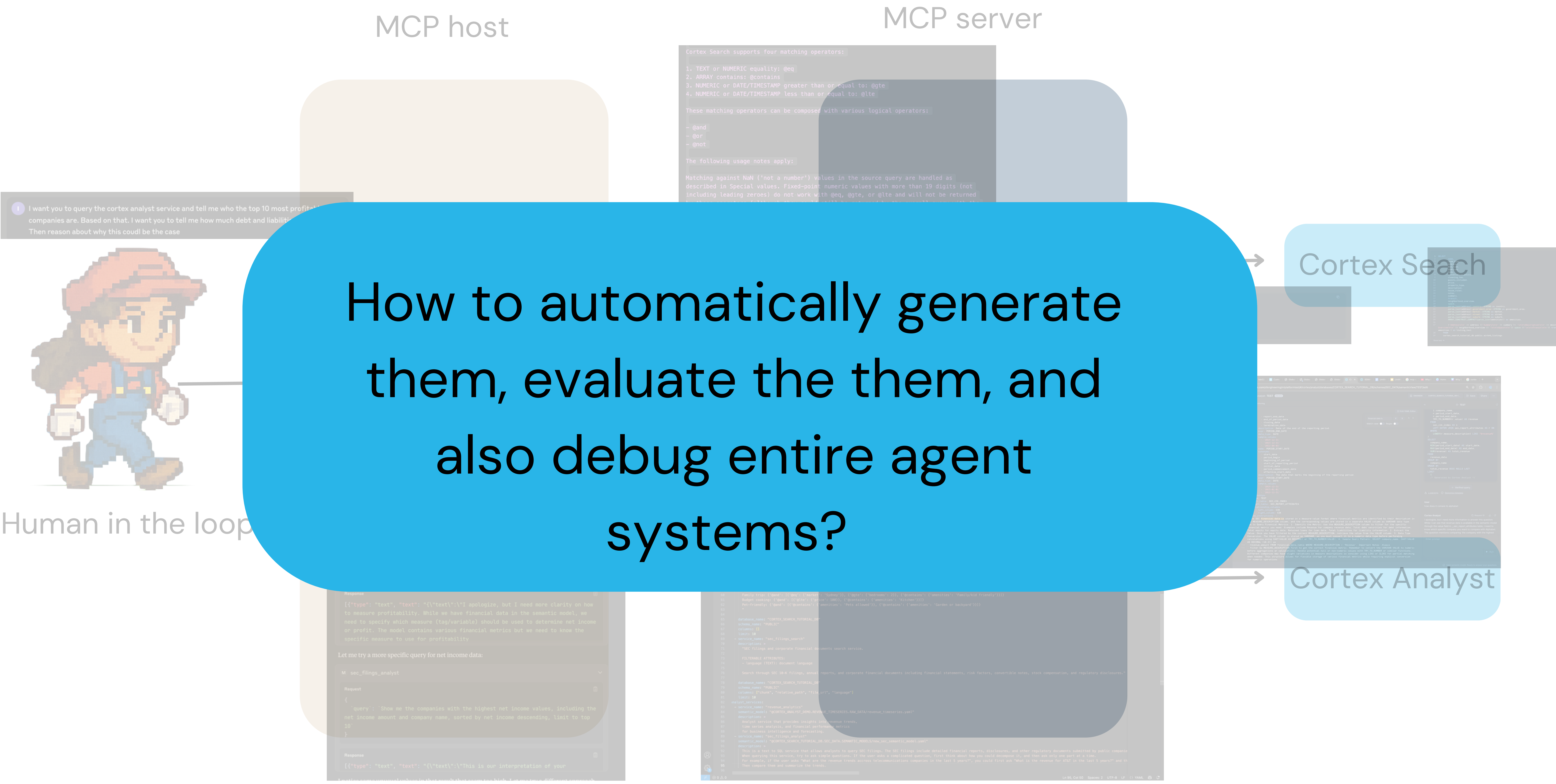
Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

How to automatically generate them, evaluate the them, and also debug entire agent systems?



I want you to query the cortex analyst service and tell me who the top 10 most profita
companies are. Based on that. I want you to tell me how much debt and liabiliti
Then reason about why this coudl be the case

```
Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as
described in Special values. Fixed-point numeric values with more than 19 digits (not
including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Response

```
[{"type": "text", "text": "{\"text\":\"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability
```

Let me try a more specific query for net income data:

sec_filings_analyst

Request

```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\":\"This is our interpretation of your
```

```
      Family trip: {'@and': [{'@eq': {'market': 'Sydney'}}, {'@gte': {'bedrooms': 2}}, {'@contains': {'amenities': 'Family/kid friendly'}}]}
      Budget cooking: {'@and': [{'@lte': {'price': 100}}, {'@contains': {'amenities': 'Kitchen'}}]}
      Pet-friendly: {'@and': [{'@contains': {'amenities': 'Pets allowed'}}, {'@contains': {'amenities': 'Garden or backyard'}}]}
      "

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: []
    limit: 10
  - service_name: "sec_filings_search"
    description: >
      "SEC filings and corporate financial documents search service.

      FILTERABLE ATTRIBUTES:
      - language (TEXT): document language

      Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."

    database_name: "CORTEX_SEARCH_TUTORIAL_DB"
    schema_name: "PUBLIC"
    columns: ["chunk", "relative_path", "file_url", "language"]
    limit: 10
nalyst_services:
  - service_name: "revenue_analytics"
    semantic_model: "@CORTEX_ANALYST_DEMO.REVENUE_TIMESERIES.RAW_DATA/revenue_timeseries.yaml"
    description: >
      Analyst service that provides insights into revenue trends,
      time series analysis, and financial performance metrics
      for business intelligence and forecasting.
  - service_name: "sec_filings_analyst"
    semantic_model: "@CORTEX_SEARCH_TUTORIAL_DB.SEC_DATA.SEMANTIC_MODELS/new_sec_semantic_model.yaml"
    description: >
      This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companie
      When querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time.
      For example, if the user asks "What are the revenue trends accross telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and th
      Then compare them and summarize the trends.
```

Ln 95, Col 50 Spaces: 2 UTF-8 LF YAML

WHAT DID I DO THIS SUMMER?

MCP Evals! → Dataset Generation