1. **(Baseline Tool for Every Task) Device & System Control** *This group is for interacting with the user's device settings.*
    - `get_system_settings`: Retrieves a summary of all system settings.
    - `set_system_settings`: Modifies multiple system settings at once.
    - *Specific Getters*: `get_cellular_status`, `get_wifi_status`, `get_location_service_status`, `get_low_battery_mode_status`.
    - *Specific Setters*: `set_cellular_status`, `set_wifi_status`, `set_location_service_status`, `set_low_battery_mode_status`, `set_locale`.
2. **Location & Navigation** *This group is for finding, understanding, and navigating to physical places.*
    - `search_place`: Finds places based on a text query.
    - `place_details`: Gets comprehensive information about a place using its ID.
    - `get_directions`: Calculates a route between two points.
    - `convert_place_id_lat_lon_address`: Converts between different location identifiers.
    - `search_place_and_specify_return_fields`: A version of search that lets you choose which data fields are returned.
    - `place_details_and_specify_return_fields`: A version of details that lets you choose which data fields are returned.
    - *Specific Search Tools*: `search_place_and_return_address`, `search_place_and_return_name`, `search_place_and_return_place_id`, `search_place_and_return_geometry`.
    - *Specific Detail Tools*: `place_details_and_return_geometry`, `place_details_and_return_address`, `place_details_and_return_reservable`, `place_details_and_return_phone_number`, etc.
    - `get_current_location`: Fetches the user's current physical location.
    - `get_current_location_field`: Gets a specific piece of location data (e.g., just the latitude).
3. **Calendar & Productivity** *This group focuses on managing schedules, events, and personal organization.*
    - `create_calendar`: Makes a new calendar (e.g., "Work," "Personal").
    - `search_calendars`: Finds existing calendars by title or ID.
    - `modify_calendar`: Edits a calendar's title.
    - `remove_calendar`: Deletes a calendar and all its events.
    - `create_calendar_event`: Adds a new event to a calendar.

    - `search_calendar_events`: Finds events based on time, attendees, location, etc.
    - `modify_calendar_event`: Edits details of an existing event.
    - `remove_calendar_event`: Deletes a calendar event.
4. **Information & Web Search** *This group is for retrieving information from the web and other external sources.*
    - `web_search`: Performs a general web search, like using Google.
    - `scrape_webpage`: Extracts the text content from a specific URL.
    - `weather_forecast`: Fetches weather information for a location.
    - `search_events`: Finds local events like concerts or festivals.
5. **Entertainment & Media (Spotify)** *This group is dedicated to finding and exploring content on Spotify.*
    - `search_spotify_top_results`: A general search across all Spotify content types.
    - *Specific Search Tools*: `search_spotify_albums`, `search_spotify_artists`, `search_spotify_tracks`, `search_spotify_playlists`, `search_spotify_podcasts`, `search_spotify_genres, search_spotify_episodes`
    - *Specific Detail Tools*: `spotify_album_details`, `spotify_artist_profile`, `spotify_track_credits`, `spotify_playlist_tracks`, etc.
6. **Time & Date** *This group is for handling time-related queries.*
    - `get_current_iso_8601_datetime_with_utc_offset`: Gets the current time in a standardized format.
    - `iso_8601_datetime_with_utc_offset_to_iso_weekday`: Converts a date into the day of the week.
7. **Finance & Investing** *This group is for researching securities, tracking market data, and getting financial news.*
    - `get_market_tickers`: Lists available symbols for stocks, ETFs, or mutual funds.
    - `search_securities`: Finds a security by its name or ticker symbol (e.g., "Apple" -> "AAPL").
    - `get_market_quotes`/`get_market_quotes_snapshots`: Retrieves real-time price quotes for one or more securities.
    - `get_stock_history`: Gets historical price and volume data for a security.
    - `screen_stocks`: Finds stocks that meet certain criteria (e.g., "day_gainers," "undervalued_growth_stocks").
    - `get_stock_modules`: Provides deep-dive information on a company, such as financial statements (`income-statement`), ownership data (`institution-ownership`), or SEC filings.

    - `get_insider_trades`: Shows recent trading activity by company insiders.
    - `get_market_news`: Fetches the latest news articles related to the market or specific tickers.
8. **E-commerce & Local Business** *This group combines tools for online shopping (Amazon) and interacting with local businesses (Yelp).*
    - **Amazon Tools:**
        - `product_search`: Searches for products on Amazon with various filters like price, category, and rating.
        - `product_details`: Gets specific information for a product using its ASIN.
        - `product_reviews`: Retrieves customer reviews for a product.
        - `seller_profile`: Gets information about a specific Amazon seller.
        - `seller_reviews`: Retrieves feedback and ratings for a seller.
        - `seller_products`: Lists products sold by a particular seller.
        - `get_product_categories`: Lists the available shopping categories on Amazon.
    - **Yelp Tools:**
        - `business_details`: Gets detailed information (address, phone, hours) for a business on Yelp.
        - `business_reviews`: Retrieves customer reviews for a Yelp business.
        - `business_menu`: Fetches the menu for a restaurant, if available.
        - `business_popular_dishes`: Lists the most popular dishes at a restaurant based on reviews.
        - `search_yelp`: Searches for businesses in yelp with basic info.
9. **Social Media & Community (Reddit)** *This group is for tapping into discussions, opinions, and content on the Reddit platform.*
    - `search_reddit`: Performs a keyword search across all of Reddit or within a specific community (`subreddit`). Can filter by posts, comments, media, etc.
    - `get_subreddit_posts`: Retrieves the latest or top posts from a specific subreddit (e.g., `r/technology`).
    - `get_post_details`: Fetches the full content of a specific Reddit post, including all of its comments.
10. **News & Information** *This group is for fetching news content, either by general category or by specific keyword.*
    - **Topical News Feeds:**
        - `get_latest_news`: Gets breaking news across all categories.
        - `get_world_news`: Focuses on global events.
        - `get_business_news`: For financial and corporate news.
        - `get_entertainment_news`: For news on movies, music, and pop culture.
        - `get_health_news`: For medical and wellness updates.

        - `get_science_news`: For scientific discoveries and research.
        - `get_sport_news`: For sports headlines and scores.
        - `get_technology_news`: For news on gadgets, software, and the tech industry.
    - **Search-Based News:**
        - `search_news`: Finds news articles based on a specific keyword or topic.
        - `get_search_suggestions`: Provides autocomplete ideas for a news search.
11. **Holidays & Culture** *This group is for identifying public holidays and observances in different countries.*
    - `get_country_holidays_by_year`: Retrieves a list of all official public holidays for a specific country and year.
    - `get_supported_country_codes`: Provides a list of all countries for which holiday data is available.

Tool Collections & Example Trajectories

These collections represent "personas" or common use cases, showing how multiple tools can be chained together to fulfill complex user requests.

Collection 1: Local Guide / Navigator

This collection is designed for tasks related to exploring the physical world, finding interesting places, and navigating between them. It's the digital equivalent of a travel guide and a GPS.

Primary Tools: `search_place`, `place_details`, `get_directions`, `weather_forecast`, `get_current_location`.

Example Trajectory:

- **User Query:** "Find me a well-rated ramen restaurant in Greenwich Village that's open now, and tell me how to walk there from my current location. Also, what will the weather be like?"

- **Generated Trajectory (Sequence of Tool Calls):**

    1. `get_current_location()` -> To establish the starting point for directions.
    2. `get_current_iso_8601_datetime_with_utc_offset()` -> To determine the current time to check if a restaurant is open.

3. `search_place(query="ramen in Greenwich Village")` -> Returns a list of potential ramen restaurants with their `place_id`s.
4. *(Internal Logic: The system iterates through the results from step 3)* `place_details(place_id="<place_id_from_step_3>")` -> Checks the `rating` and `opening_hours` for each place to find one that is highly-rated and currently open.
5. *(Once a suitable restaurant is found)* `get_directions(origin="<place_id_from_step_1>", destination="<restaurant_place_id>", mode="walking")` -> To generate the walking route.
6. `weather_forecast(latlng="<current_latitude,longitude>")` -> To get the weather for the user's walk.
7. **Final Output:** The system synthesizes the information from steps 4, 5, and 6 into a coherent answer: "I found a place called 'Ippudo' with a 4.5-star rating. It's open now. Here are the walking directions from your location; it should take about 15 minutes. The weather is clear, around 68°F."

Collection 2: Proactive Personal Assistant

This collection focuses on managing a user's life, including their schedule, device settings, and personal reminders. It acts like an executive assistant that can handle complex, multi-part instructions.

Primary Tools: All `Calendar & Productivity` tools, all `Device & System Control` tools, and all `Time & Date` tools.

Example Trajectory:

- **User Query:** "I have a flight to San Francisco next Tuesday. Please add a 2-hour 'Pack for trip' event to my 'Personal' calendar for next Monday evening. Also, make sure my phone is not in low battery mode that day."

- **Generated Trajectory (Sequence of Tool Calls):**

1. `get_current_iso_8601_datetime_with_utc_offset()` -> To understand "next Tuesday" and "next Monday" relative to today.
2. `search_calendars(title="Personal")` -> To find the `calendar_id` for the user's "Personal" calendar.
3. *(Internal Logic: The system calculates the date and time for "next Monday evening," for example, 7:00 PM)*.

4. `Calendar(calendar_id="<id_from_step_2>", title="Pack for trip", start_datetime="2025-06-23T19:00:00-04:00", end_datetime="2025-06-23T21:00:00-04:00")` -> To create the event.
5. `set_low_battery_mode_status(status=False)` -> To fulfill the second part of the request.
6. **Final Output:** The system confirms the action: "Okay, I've added a 'Pack for trip' event to your Personal calendar for next Monday from 7 PM to 9 PM. I've also made sure low battery mode is turned off."

Collection 3: Entertainment / Media Explorer

This collection is for users looking to discover and engage with media, find events, and learn more about artists and their work.

Primary Tools: All `Spotify` tools, `search_events`, `web_search`, `scrape_webpage`.

Example Trajectory:

- **User Query:** "I just heard a song by an artist named 'Khruangbin' on the radio. Find their most popular album on Spotify. What are the top 3 songs on it? Also, find out what their name means."

- **Generated Trajectory (Sequence of Tool Calls):**

1. `search_spotify_artists(q="Khruangbin")` -> To get the artist's unique Spotify `artist_id`.
2. `spotify_artist_discography(artist_id="<id_from_step_1>")` -> To find all their albums.
3. *(Internal Logic: The system would need to determine the "most popular" album, perhaps by looking at the release date or searching for it on the web.)* Let's assume it identifies an album.
4. `spotify_album_details(ids="<album_id>")` -> To get the list of tracks on that album and their popularity metrics. The system then picks the top 3.
5. `web_search(query="what does Khruangbin mean")` -> To find the meaning of the band's name.
6. `scrape_webpage(url="<url_from_search_result>")` -> To extract the answer from a reliable source like Wikipedia or an interview.
7. **Final Output:** The system combines the information: "Khruangbin's most popular album is 'Con Todo El Mundo.' The top three tracks are 'Maria También,' 'August 10,' and 'Evan Finds the Third Room.' The name 'Khruangbin' is Thai for 'airplane' (เครื่องบิน)."

Of course. Let's break down the finance, e-commerce, and Reddit tools using the same format.

Collection 4: Investor

This collection is for researching investments, analyzing performance, and staying up-to-date on market news. It helps users move from a general idea to specific, data-backed insights.

Primary Tools: All `Finance & Investing` tools.

Example Trajectory:

- **User Query:** "I'm thinking about investing in cloud computing. How has Google's stock performed over the last six months, and what is jejich latest news? Also, show me their recent revenue from their last income statement."
- **Generated Trajectory (Sequence of Tool Calls):**
    1. `search_securities(search="Google")` -> To find the official ticker symbol, which is `GOOGL`.
    2. `get_stock_history(symbol="GOOGL", interval="1d")` -> The system would then process this data to show the performance over the last 6 months.
    3. `get_market_news(tickers="GOOGL")` -> To fetch the latest articles and headlines related to Google.
    4. `get_stock_modules(ticker="GOOGL", module="income-statement")` -> To retrieve the company's income statement.
    5. *(Internal Logic: The system parses the income statement data to find the "totalRevenue" figure for the most recent period.)*
    6. **Final Output:** The system synthesizes all the retrieved data: "Over the last six months, Google's stock (GOOGL) has seen a 12% increase. Recent news includes their announcement of a new AI model... From their latest income statement, their quarterly revenue was $86.31 billion."

Collection 5: eCommerce/shopping

This user wants to find the best product for their needs, relying on reviews, price comparisons, and other details before making a decision. This trajectory blends online e-commerce with local business information.

Primary Tools: All `E-commerce & Local Business` tools (Amazon & Yelp).

Example Trajectory:

- **User Query:** "I need a great gift for a friend who loves to cook. Find me a highly-rated Dutch oven on Amazon. For the top result, what do the reviews say about its durability? Then, find a local kitchen supply store in NYC that might sell it, and tell me what their most popular items are."

- **Generated Trajectory (Sequence of Tool Calls):**

    1. `product_search(query="dutch oven", sort_by="REVIEWS", four_stars_and_up=True)` -> To find the top-rated Dutch ovens on Amazon.
    2. *(Internal Logic: The system identifies the top product from the results, let's say it's a Le Creuset model with a specific ASIN.)*
    3. `product_reviews(asin="<le_creuset_asin>", query="durability")` -> To specifically find reviews that mention the product's durability.
    4. `search_place(query="kitchen supply store in New York")` -> To find local stores that might carry the item. The tool `search_place` from the *other* toolset is used here.
    5. *(Internal Logic: The system picks the top local store result, e.g., "Sur La Table," and gets its `business_id` from Yelp if available, or just uses the name.)* Let's assume it finds a Yelp ID.
    6. `business_popular_dishes(business_id="<sur_la_table_yelp_id>")` -> This is a creative use of the tool. While intended for restaurants, it might return popular *products* mentioned in reviews. If that fails, `business_reviews` could be used to parse for product names.
    7. **Final Output:** The system provides a multi-part answer: "The top-rated Dutch oven on Amazon is the Le Creuset 5.5-Quart model. Reviews frequently praise its long-lasting enamel and sturdy build. Locally, you could check 'Sur La Table' in SoHo. While I can't check their live stock, reviews on their Yelp page often mention their high-quality knives and bakeware as popular items."

Collection 6: Community Researcher

This user leverages the "wisdom of the crowd" on Reddit to get authentic opinions, advice, and solutions to problems that require real-world experience.

Primary Tools: All `Reddit` tools.

Example Trajectory:

- **User Query:** "My dog gets really anxious during thunderstorms. What do people on the `r/dogs` subreddit recommend for calming anxious dogs?"

- **Generated Trajectory (Sequence of Tool Calls):**

    1. `search_reddit(search="anxious during thunderstorms", subreddit="dogs", sort_type="comments")` -> This searches for posts where this topic has generated the most discussion, indicating high engagement and multiple viewpoints.
    2. *(Internal Logic: The system scans the titles and snippets of the top 5-10 search results to find the most relevant threads.)*
    3. `get_post_details(url="<most_relevant_post_url>")` -> The system selects the most promising post and fetches all its comments.
    4. *(Internal Logic: The system analyzes the text of the comments, looking for recurring suggestions, product names, or techniques. It might perform entity recognition to find things like "Thundershirt," "calming treats," "white noise machine," or "positive reinforcement.")*
    5. **Final Output:** The system summarizes the most common advice found in the community discussion: "On the r/dogs subreddit, several common recommendations for thunderstorm anxiety came up. Many users have had success with a 'Thundershirt' or similar anxiety wrap. Others suggest using calming treats with ingredients like melatonin or L-theanine. Creating a 'safe space' for the dog with blankets and playing white noise to drown out the thunder are also frequently mentioned."

Collection 7: News

This user wants to stay on top of the news, whether it's general headlines or deep dives into specific topics that interest them. This collection allows for both broad awareness and targeted research.

Primary Tools: All `News & Information` tools.

Example Trajectory:

- **User Query:** "What are today's top headlines in technology? Also, I'm interested in the future of autonomous vehicles; search for recent news about that."

- **Generated Trajectory (Sequence of Tool Calls):**

    1. `get_technology_news()` -> To fetch the general, top-level technology news for the day.

2. `search_news(keyword="autonomous vehicle technology")` -> To perform a focused search on the user's specific interest.
3. *(Internal Logic: The system processes the results from both calls, potentially filtering out any duplicate stories to provide a clean, combined summary.)*
4. **Final Output:** The system presents a two-part answer: "Today's top technology headlines include Amazon's new robotaxi factory and Meta's rollout of Passkey logins. Regarding autonomous vehicles, Zoox (owned by Amazon) is expanding its robotaxi service to Las Vegas and San Francisco, aiming to challenge Waymo's dominance in the field."

Collection 8: Global Planner

This user is organizing international projects, travel, or events and needs to be mindful of local schedules and cultural observances. This collection helps avoid scheduling conflicts with public holidays.

Primary Tools: `get_country_holidays_by_year`, `get_supported_country_codes`, potentially paired with `Calendar` tools.

Example Trajectory:

- **User Query:** "I need to schedule a series of virtual workshops with our teams in the United States, Germany, and Japan for the first week of October this year. Are there any public holidays in those countries I should be aware of?"

- **Generated Trajectory (Sequence of Tool Calls):**

1. *(Internal Logic: The system identifies the current year (2025) and the correct ISO codes for the countries: "US", "DE", and "JP".)*
2. `get_country_holidays_by_year(country_code="US", year=2025)`
3. `get_country_holidays_by_year(country_code="DE", year=2025)`
4. `get_country_holidays_by_year(country_code="JP", year=2025)`
5. *(Internal Logic: The system takes the lists of holidays from all three calls and filters them to find any that fall within the first week of October 2025.)*
6. **Final Output:** The system provides a direct, actionable answer: "Yes, you should be aware of one major holiday. Friday, October 3rd, 2025, is German Unity Day, a national holiday in Germany. The US and Japan have no national holidays during that specific week, though Sports Day in Japan is the following Monday, October 13th."