System Architecture & Agents Planning

Tooling & Technology Stack
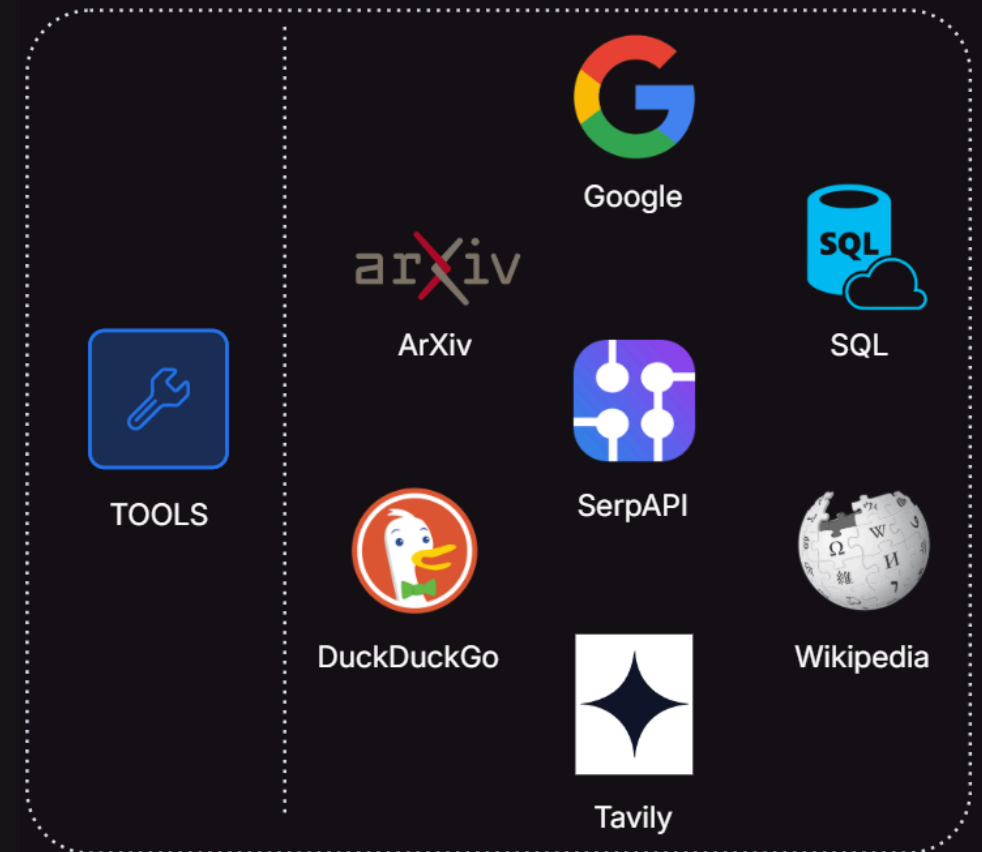
Nitin Agarwal

Principal Data Scientist

Ex-Microsoft

Understanding Tools for Agents

- Tools are **callable Python functions or APIs** that agents invoke to fetch or compute data.

- They act as **"hands" of the agent,** performing real-world actions or fetching data that the LLM cannot generate on its own.

- Each tool encapsulates a specific function, such as fetching stock information, calculating indicators, or assessing risk.

TOOLS

Google

arXiv — ArXiv

SQL

SerpAPI

DuckDuckGo

Tavily

Wikipedia

Analytics Vidhya

Why Designing Tools is Important?
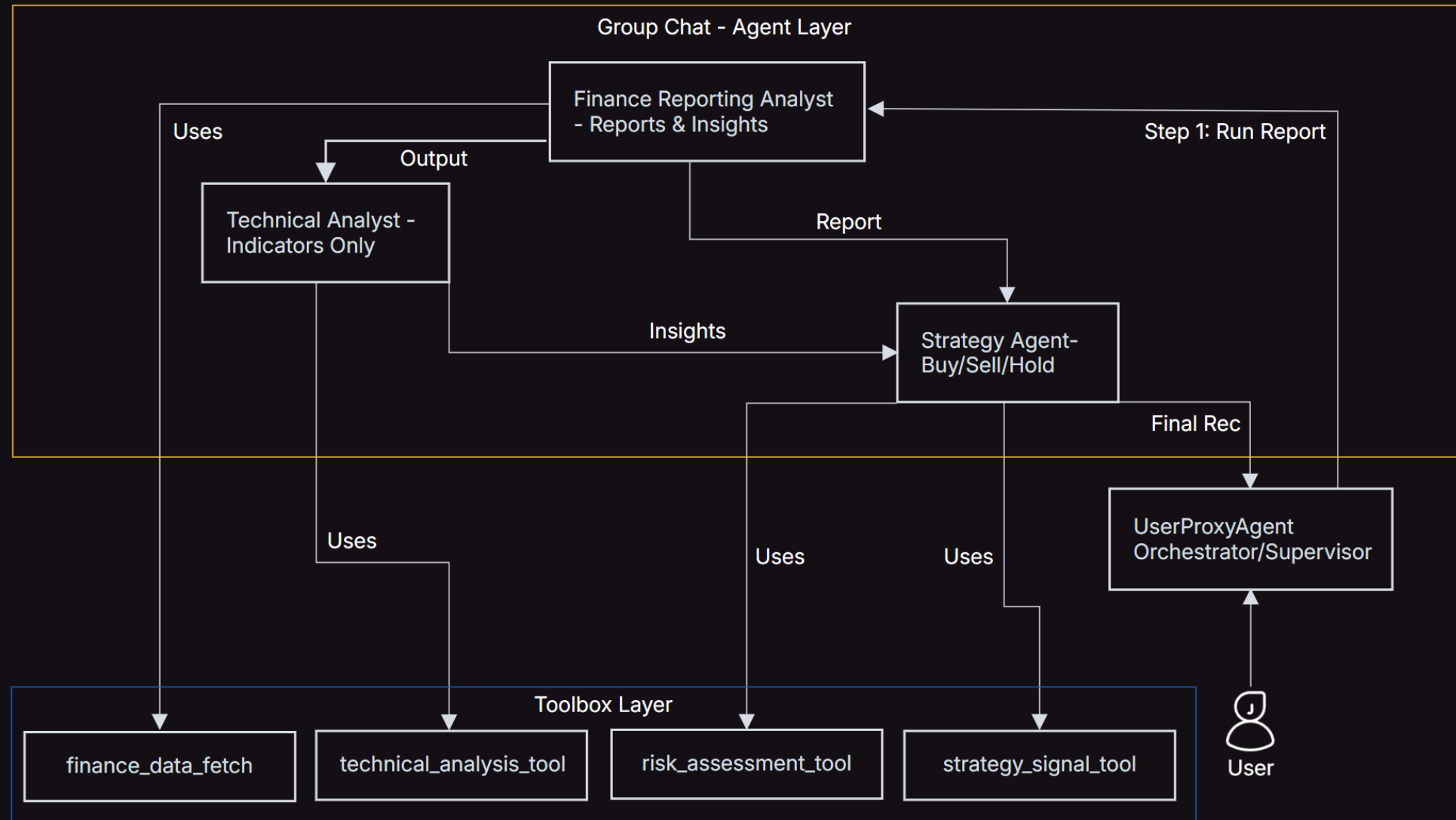

Modularity & Accuracy

Tools separate logic from execution, ensuring reliable, verifiable, and efficient operations while minimizing hallucinations.

Reusability & Scalability

Well-designed tools are easy to extend, reuse across agents, and help maintain safety and control in complex workflows.

Analytics Vidhya

Agents and Tools for Financial Analysis

Group Chat - Agent Layer

Finance Reporting Analyst - Reports & Insights

Step 1: Run Report

Uses

Output

Technical Analyst - Indicators Only

Report

Insights

Strategy Agent- Buy/Sell/Hold

Final Rec

UserProxyAgent Orchestrator/Supervisor

Uses

Uses

Uses

Toolbox Layer

finance_data_fetch

technical_analysis_tool

risk_assessment_tool

strategy_signal_tool

User

Analytics Vidhya

Tool 1: finance_data_fetch

- **Purpose:** Fetch basic financial and historical stock data

- **Inputs:** stock ticker, period (optional, default = 1mo)

- **Metrics Extracted:**
  - Company info (name, market cap, price, ratios)
  - Historical closing prices

- **Used by:** finance_reporting_analyst

- **Error Handling:** Graceful fallback with error message in JSON

```
{
  "name": "finance_data_fetch",
  "data": {
    "name": "Apple Inc.",
    "symbol": "AAPL",
    "currentPrice": 192.32,
    "currency": "USD",
    "summary": "Apple Inc. designs, manufactures,
and markets smartphones, personal computers,
tablets...",
    "marketCap": 3000000000000,
    "peRatio": 29.87,
    "priceToBook": 45.3,
    "dividend": 0.92,
    "recentClosePrices": {
      "2024-06-01": 189.50,
      "2024-06-02": 190.40,
      "2024-06-03": 191.70,
      "2024-06-04": 192.32
    }
  }
}
```

Analytics Vidhya

Tool 2: technical_analysis_tool

- **Purpose:** Perform basic technical indicator calculations

- **Metrics Extracted:**
  - SMA (Simple Moving Avg) – 20-day
  - EMA (Exponential Moving Avg) – 20-day
  - RSI (Relative Strength Index) – 14-day
  - Last close price

- **Used by:** technical_analyst to detect trends and momentum

- **Error Handling:** Graceful fallback with error message in JSON

```json
{
    "name": "technical_analysis_tool",
    "data": {
        "SMA_20": 188.45,
        "EMA_20": 190.21,
        "RSI": 62.8,
        "Last_Close": 192.32
    }
}
```

Analytics Vidhya

Tool 3: risk_assessment_tool

- **Purpose:** Analyze the stock's risk profile

- **Metrics Extracted:**
  - Beta (volatility)
  - Market Cap
  - Dividend Yield
  - 52-week Change (Volatility proxy)
  - Risk Rating (High / Moderate / Low)

- **Used by:** strategy_agent for risk-informed recommendations

- **Error Handling:** Graceful fallback with error message in JSON

```json
{
  "name":
"risk_assessment_tool",
  "data": {
    "Beta": 1.15,
    "MarketCap":
300000000000,
    "DividendYield": 0.005,
    "Volatility": 0.18,
    "RiskRating": "Moderate"
  }
}
```

Analytics Vidhya

Tool 4: strategy_signal_tool

- **Purpose:** Buy/Sell Signals - Support decision-making

- **Indicators Used:**
  - MACD (Momentum)
  - MACD Signal Line
  - RSI
  - Last close price

- **Used by:** strategy_agent for buy/sell/hold recommendations

- **Error Handling:** Graceful fallback with error message in JSON

```
{
  "name":
"strategy_signal_tool",
  "data": {
    "MACD": 1.43,
    "MACD_Signal": 1.12,
    "RSI": 63.5,
    "Last_Close": 192.32
  }
}
```

Analytics Vidhya

Thanks