

Symfony & Agentic Applications

Christopher Hertel - Contao Konferenz 2025

Agenda

- Basic Concepts & Terminology
Models, Context, Platform, and more
- Evolution from LLM to Agentic
How Models Power Agentic Applications
- Model Context Protocol
Distributed Agentic Applications



Christopher Hertel

- Software Engineer
- Focus on PHP, Symfony and GenAI
- Working on Tech Program of amcs[✦]

Basic Concepts & Terminology

Models, Context, Platform, and more

Models

- Language Model

Trained to generate text based on text input

- Strong Competition & Fast Innovation

OpenAI, Google, Meta, Anthropic, Mistral, and more

- Various Capabilities

Streaming, structured output, multi-modal, real-time, tool calling

- Reasoning Model

Decision-making, less prompt engineering & more autonomous



Platforms

- **Model Vendor**
OpenAI, Anthropic, Mistral, Google, ...
- **General Cloud Provider**
Azure (GitHub), Google, AWS, ...
- **Specialized Cloud Provider**
HuggingFace, Replicate, OpenRouter, ...
- **Local Runtimes**
GPT4All, Ollama, ONNX Runtime, ...



Context

- Stateless API

Pre-trained, fine-tuned, but API "forgets"

- Limitation

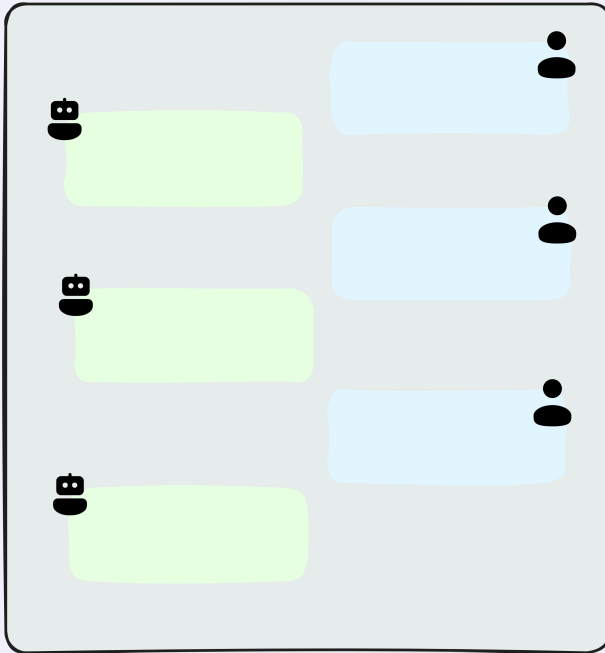
Size of context varies per model, but limited

- Powerful Control Mechanism

Efficient to steer and evaluate the models generation

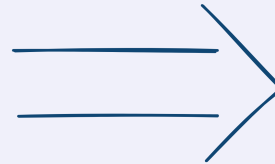
Context Example

Message Context of Chat UI

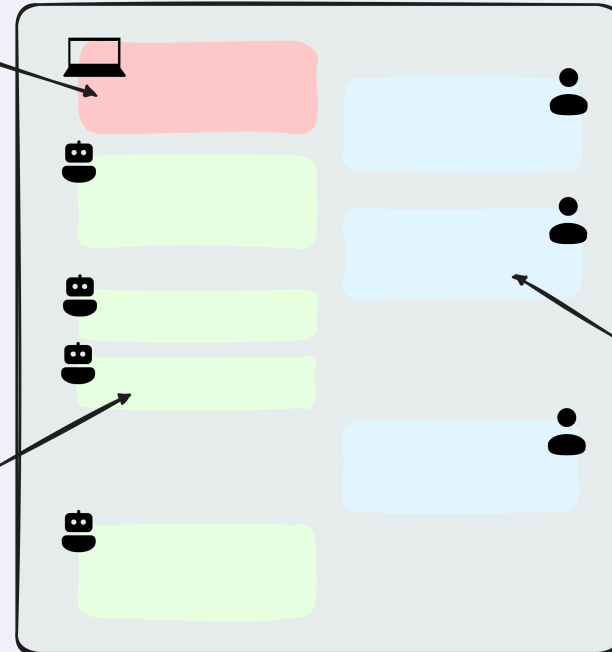


System Prompt

Extended Context of Model to work with



Additional Information



Message Manipulation

PHP Libraries

- Listicle on GitHub
- LLPhant
[theodo-group/llphant](https://github.com/theodo-group/llphant)
- Prism
[prism-php/prism](https://github.com/prism-php/prism)
- **Symfony AI**
[symfony/ai](https://github.com/symfony/ai)



Today

Symfony AI

New Symfony Initiative around AI

Symfony AI

- Published July 2025
Published, not released, still experimental - maybe EOY?
- Succeeding PHP-LLM with LLM Chain
You might not have heard of that anyways
- Standalone mono-repository
Not part of [symfony/symfony](#), but [symfony/ai](#)
- Symfony Standards will be applied
From code style to backward compatibility promise



Components

- **Platform Component**
Low-level abstraction layer for model inference
- **Store Component**
Low-level abstraction layer for vector store integrations
- **Agent Component**
High-level agentic or multi-agent framework
- **AI Bundle**
Framework integration bundle for AI components



Want to get started?

Check Demo and Examples in [symfony/ai](#) Repository

Evolution from LLM to Agent

How Models Power Agentic Applications

Install AI Platform

```
composer require symfony/ai-platform
```



Platform & Model Usage

```
use Symfony\AI\Platform\Bridge\OpenAi\{Gpt, PlatformFactory};
use Symfony\AI\Platform\Message\{Message, MessageBag};

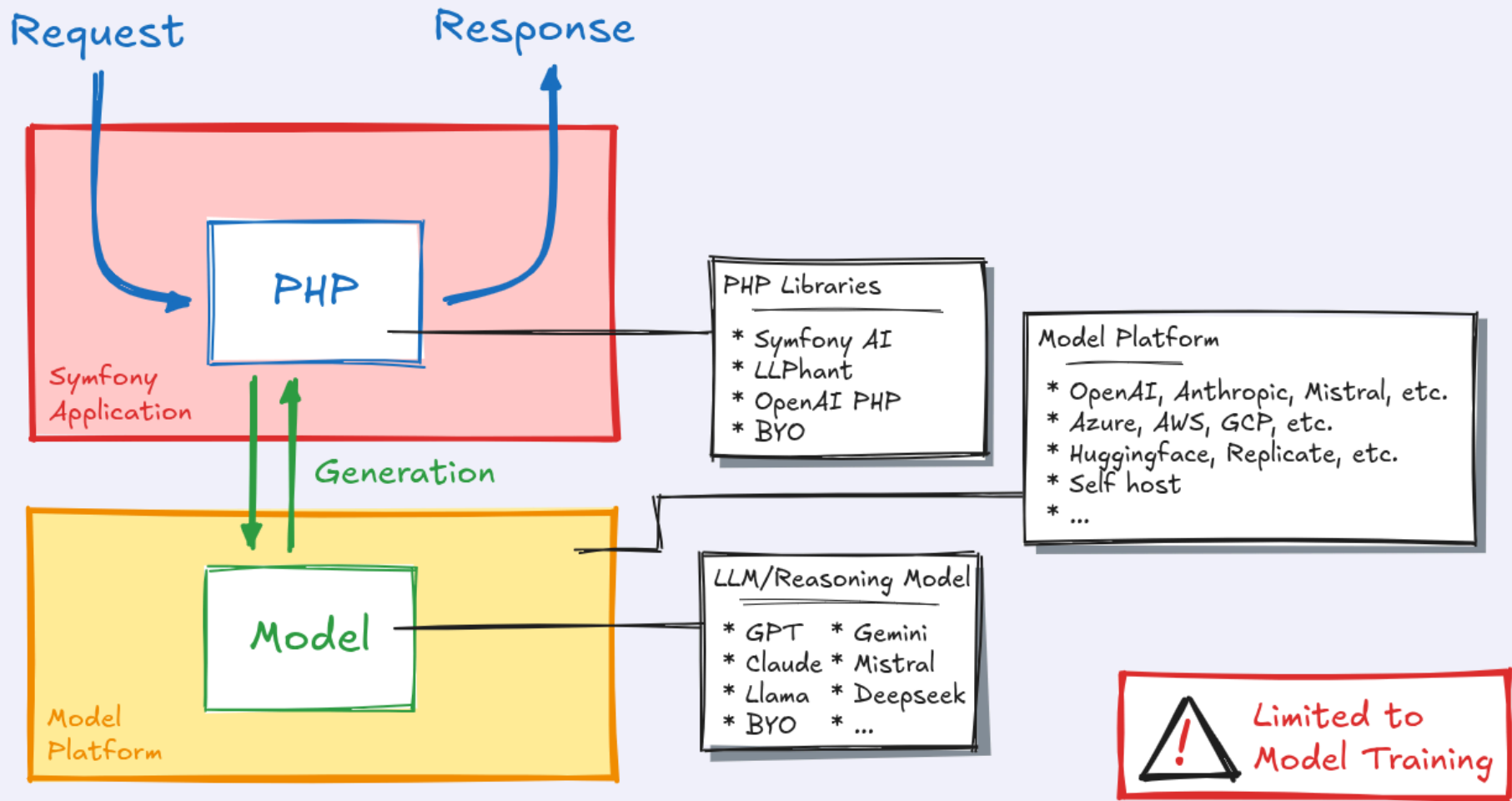
$platform = PlatformFactory::create($apiKey);
$model = new Gpt('gpt-4o-mini');

$input = new MessageBag(
    Message::forSystem('You are a pirate and you write funny.'),
    Message::ofUser('What is the Symfony framework?'),
);

$result = $platform->invoke($model, $input);

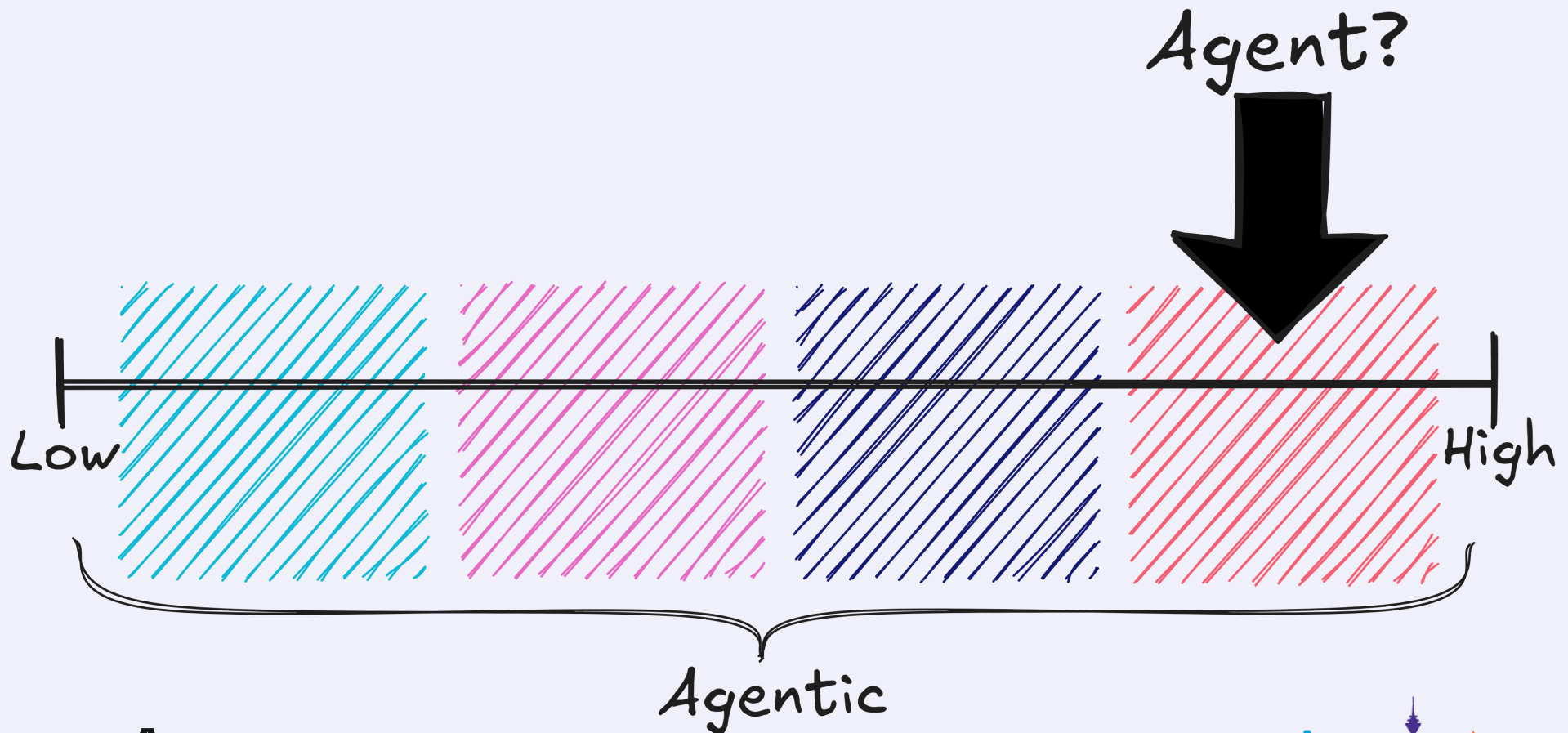
echo $result->asText();
```


Plain Model Call



Agent vs. Agentic

Level of Autonomy



Agent

- Multistep Calls

Series of calls between applications and models

- Combination of Things

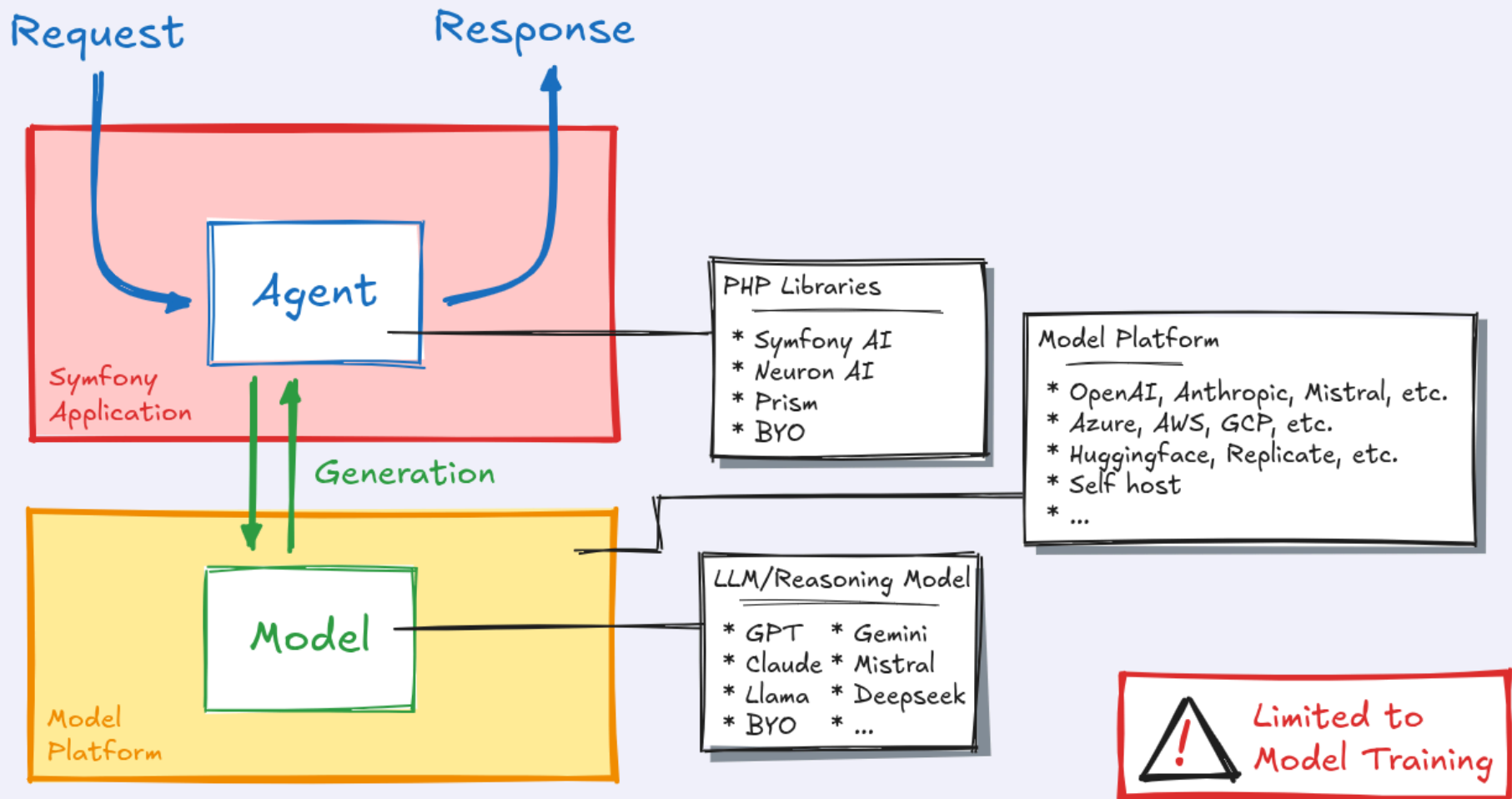
Platform, Models, System Prompts, Tools, and more

- Various Architectures

From static workflows to autonomous orchestration



Basic Agent Setup



Install AI Agent

```
composer require symfony/ai-agent
```

Basic Agent Setup

```
use Symfony\AI\Agent\Agent;  
use Symfony\AI\Platform\Bridge\OpenAi\{Gpt, PlatformFactory};  
use Symfony\AI\Platform\Message\{Message, MessageBag};  
  
$platform = PlatformFactory::create($apiKey);  
$agent = new Agent($platform, new Gpt('gpt-4o-mini'));  
  
$messages = new MessageBag(  
    Message::forSystem('You are a pirate and you write funny.'),  
    Message::ofUser('What is the Symfony framework?'),  
);  
  
$response = $agent->call($messages);  
  
echo $response->getContent();
```

Symfony Integration

```
composer require symfony/ai-bundle
```



Bundle Configuration

```
# config/packages/ai.yaml
ai:
  platform:
    openai: # or anthropic, azure, google, more to come ...
    api_key: '%env(OPENAI_API_KEY)%'
  agent:
    default:
      model:
        class: 'Symfony\AI\Platform\Bridge\OpenAi\Gpt'
        name: 'gpt-4o-mini'
      prompt: 'You are a pirate and you write funny.'
```


Usage in Symfony

```
final readonly class SomeService
{
    public function __construct(private AgentInterface $agent)
    {
    }

    public function someMethod(string $message): string
    {
        $messages = new MessageBag(Message::ofUser($message));

        $response = $this->agent->call($messages);

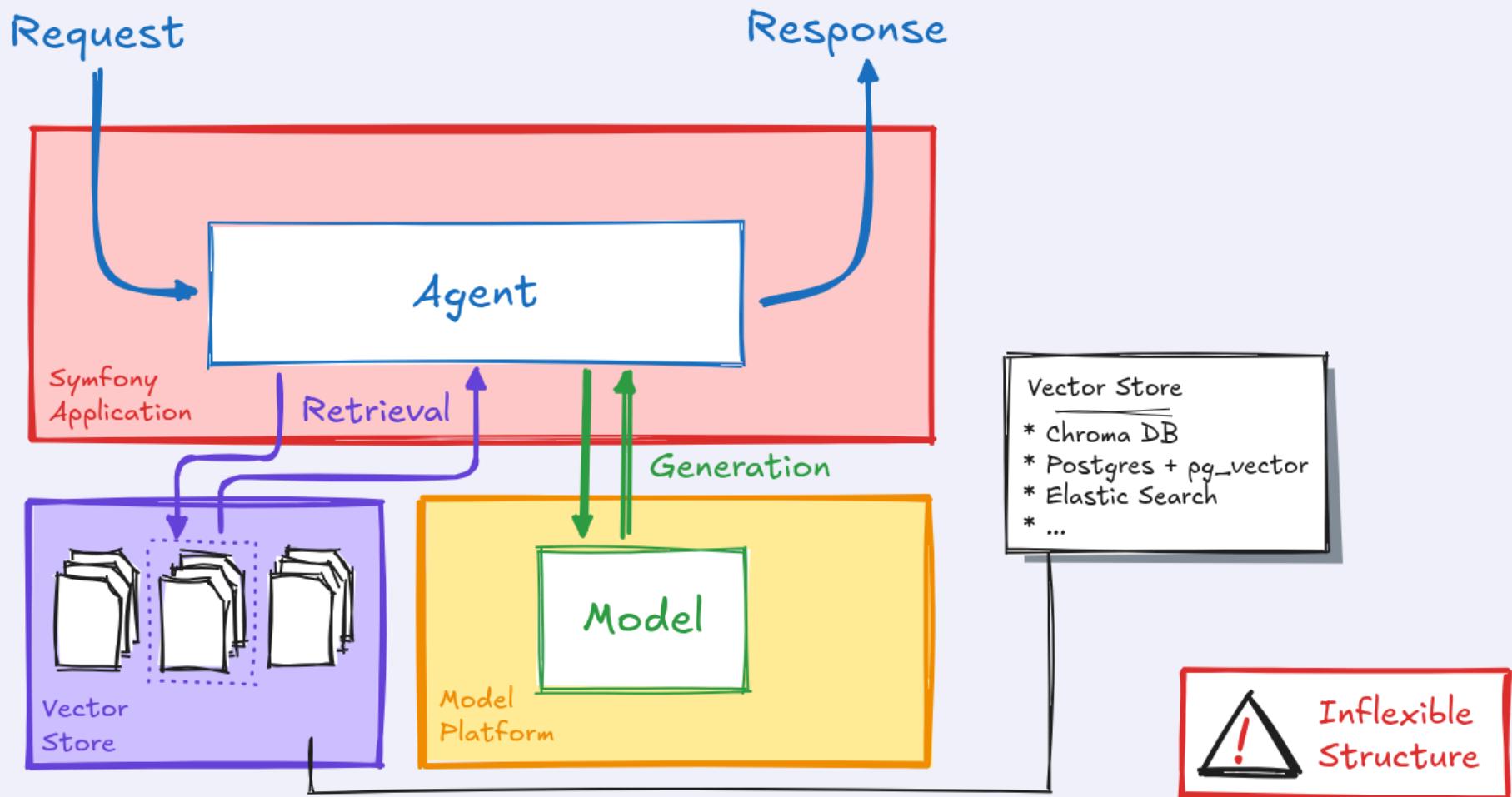
        return $response->getContent();
    }
}
```

RAG Architecture

- **Retrieval Augmented Generation**
Dynamically inject information into context
- **Improves Accuracy and Relevance**
More context for the model to work with, based on input
- **Vector Stores**
Classic RAG combines embedding vectors and similarity search
- **Improvement Techniques**
Pre-processing, Multi-Query, Reranking, and more



Simple RAG



Tool Calling

- App Provides Tools

Exposed to tool via JSON Schema definition as part of API call

- Model Invokes Tool

As a intermediate response the model invokes a tool (or multiple)

- Tool Result

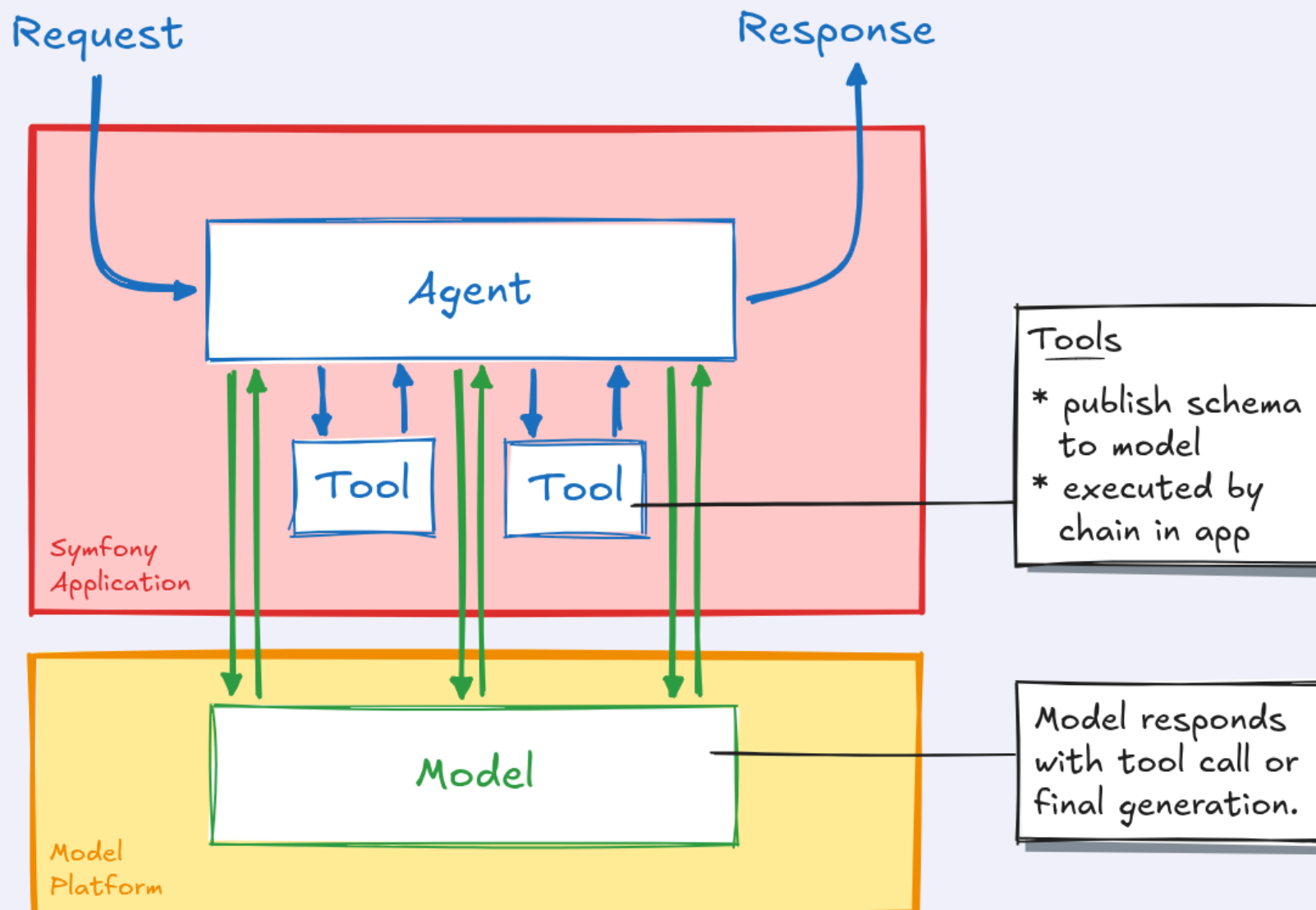
Result of tool call provided as next call to model

- Instrumentation

Log and analyze tool calls, model invocations, and results



Tool Calling



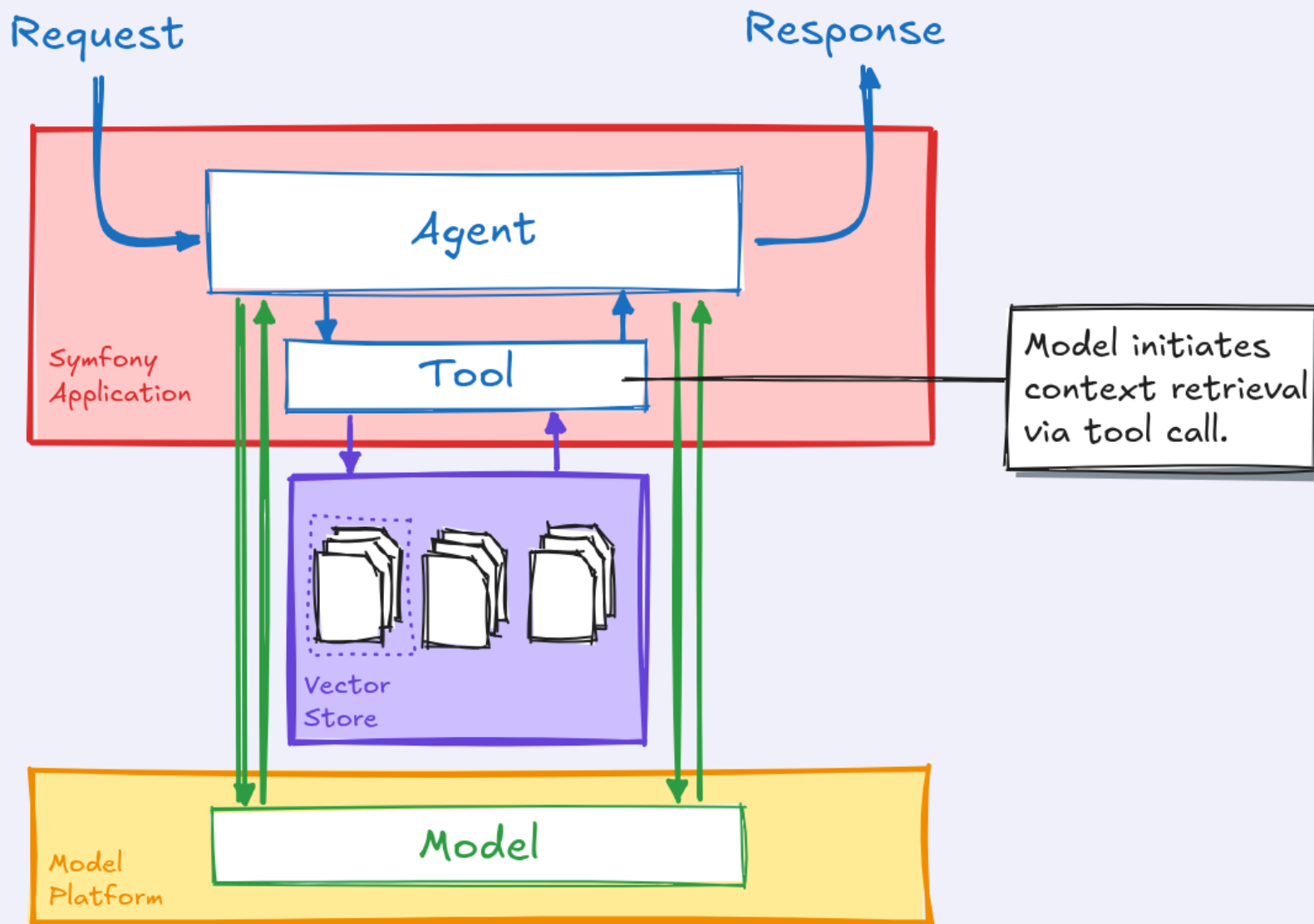
Tool Calling

```
use Symfony\AI\Agent\Toolbox\Attribute\AsTool;  
use Symfony\Component\Clock\ClockInterface;  
  
#[AsTool('clock', description: 'Returns the current date and time.')]  
final readonly class Clock  
{  
    public function __construct(private ClockInterface $clock)  
    {  
    }  
  
    public function __invoke(): string  
    {  
        return $this->clock->now()->format('Y-m-d H:i:s');  
    }  
}
```

That's it already!

Agentic RAG

31



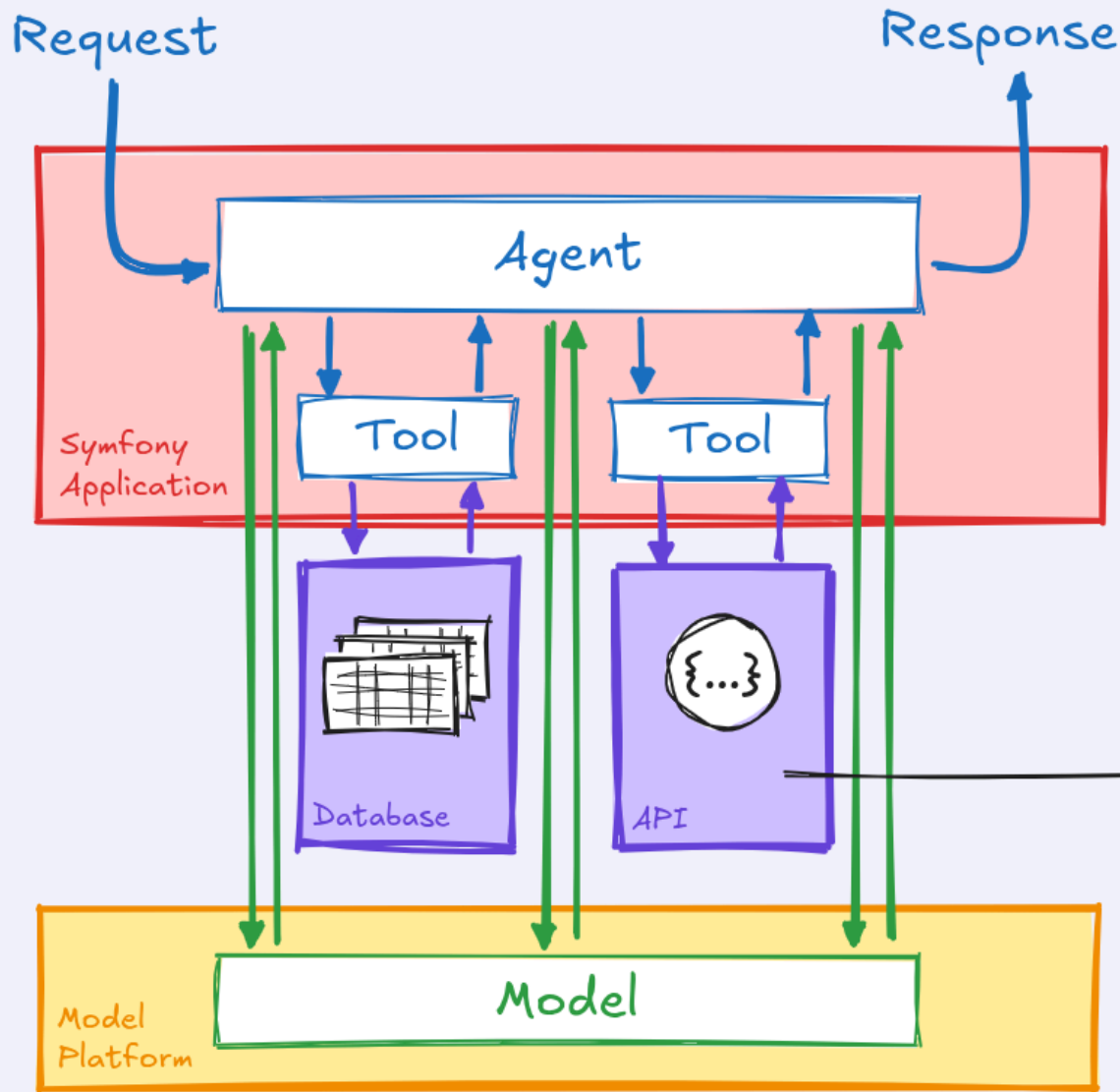
Bundle Configuration

```
# config/packages/ai.yaml
ai:
    # Platform & Agent Configuration ...
    store:
        chroma_db: # or Azure Search, MongoDB, Pinecone, more to come ...
        symfonyblog: { collection: 'symfony_blog' }
    indexer:
        default:
            model: { class: '..\Embeddings', name: 'text-embedding-ada-002' }

services:
    _defaults: { autowire: true, autoconfigure: true }

    Symfony\AI\Agent\Toolbox\Tool\SimilaritySearch: ~
```


Agentic Application

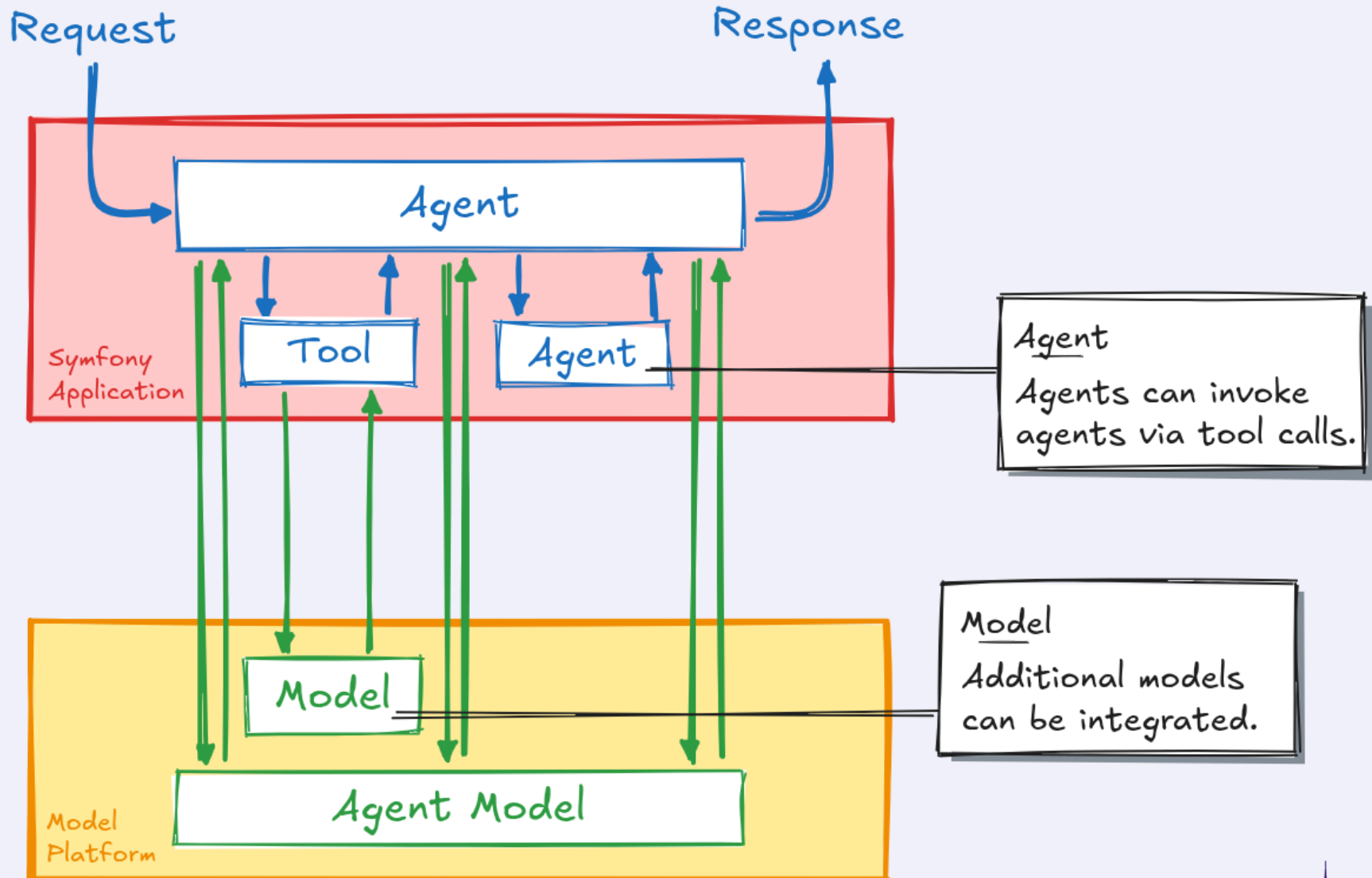


All kind of retrieval logic go into tools.

Tools can also do write operations.

Bundle Configuration

```
# config/packages/ai.yaml
ai:
  # Platform Configuration ...
  agent:
    wikipedia:
      model:
        class: 'Symfony\AI\Platform\Bridge\OpenAi\Gpt'
        name: 'gpt-4o-mini'
      prompt: |
        Please answer the users question based on information of
        Wikipedia and provide a link to the article.
    tools:
      - 'Symfony\AI\Agent\Toolbox\Tool\Wikipedia'
```



Bundle Configuration

```
# config/packages/ai.yaml
ai:
  # Platform Configuration ...
  agent:
    complex_agent:
      # Model Configuration ...
      tools:
        # Agent in agent 🤖
        - agent: 'blog'
          name: 'symfony_blog'
          description: 'Can answer questions based on the Symfony blog.'
```

Stack and isolate agents to build a more complex agents.

Model Context Protocol

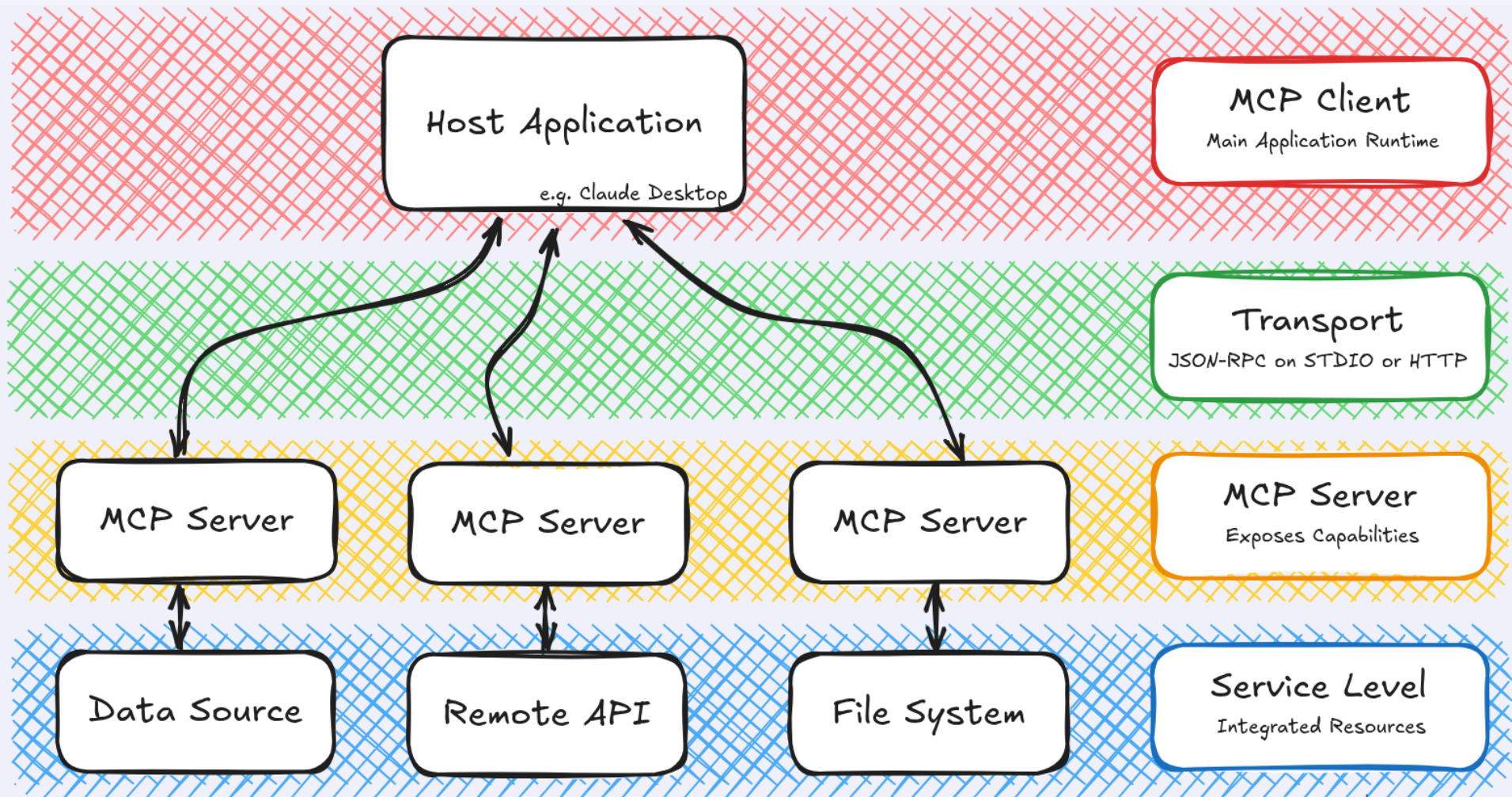
Distributed Agentic Applications

Model Context Protocol (MCP)

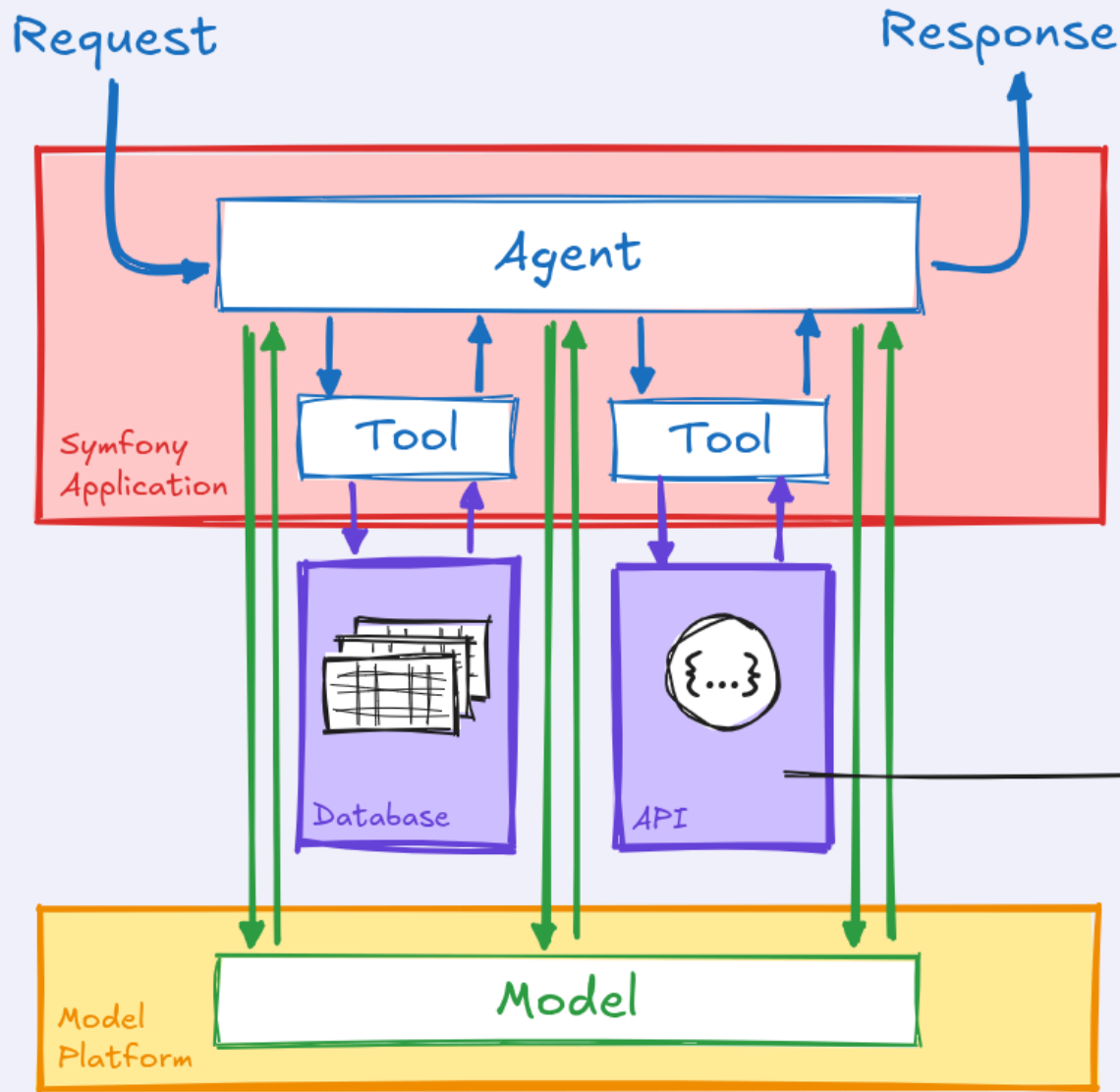
- Open Sourced Protocol by Anthropic
Only release late 2024, already exploded to everywhere
- Standardizes Five Aspects
Resources, Prompts, Tools, Sampling, Roots
- Two Main Components
Defines MCP Server and MCP Client
- JSON-RPC with Two Transports
Stdio and Streamable HTTP



General Architecture



Agentic Application



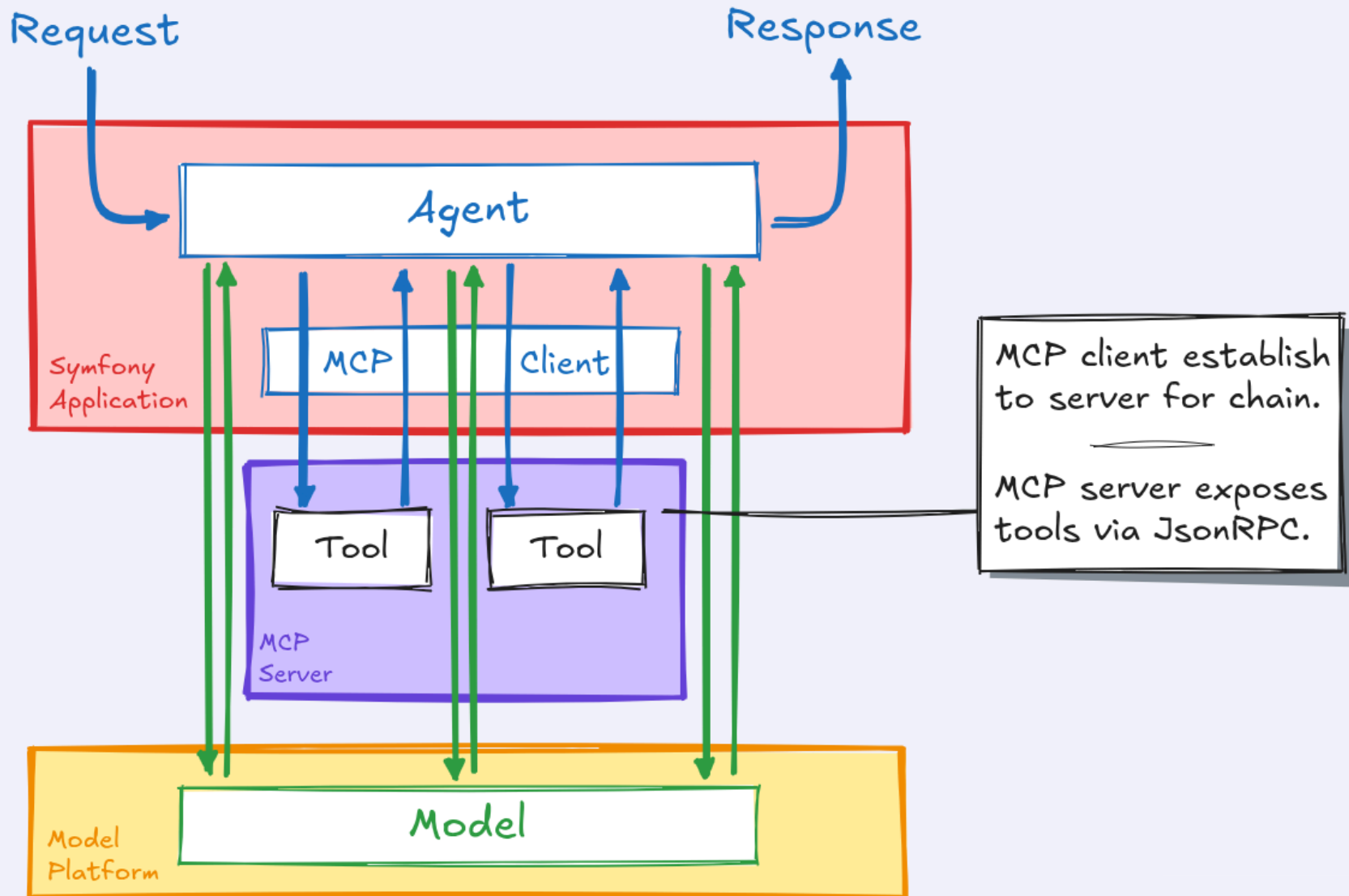
All kind of retrieval logic go into tools.

Tools can also do write operations.



Model Context Protocol

41



PHP MCP SDK

There was no SDK for PHP, but ...

Symfony teamed up

- **Symfony's MCP SDK**
Initially part of PHP-LLM and Symfony AI
- **PHP-MCP Organization**
Kyrian Obikwelu a.k.a. [CodeWithKyrian](#)
- **The PHP Foundation**
Roman Pronskiy a.k.a. [pronskiy](#)
- **Anthropic's MCP Team**
David Soria Parra a.k.a. [dsp](#)



PHP MCP SDK

```
composer require mcp/sdk
```

or

```
composer require symfony/mcp-bundle
```

Still work in progress!

Bundle Configuration

```
# config/packages/mcp.yaml
mcp:
  app: 'app' # Application name to be exposed to clients
  version: '1.0.0' # Application version to be exposed to clients

  # Configure this application to act as a MCP server
  # For now ONLY TOOLS
  client_transports:
    stdio: true # Enable STDIO via command
    http: true # Enable Server-Sent Event via controller

  # Configure MCP servers to be used by this application
  # NOT IMPLEMENTED YET
  servers:
    name:
      transport: 'stdio' # Transport method to use, either 'stdio' or 'sse'
      stdio:
        command: 'php /path/bin/console mcp' # Command to execute to start
        arguments: [] # Arguments to pass to the command
      http:
        url: 'http://localhost:8000/sse' # URL to SSE endpoint of MCP server
```

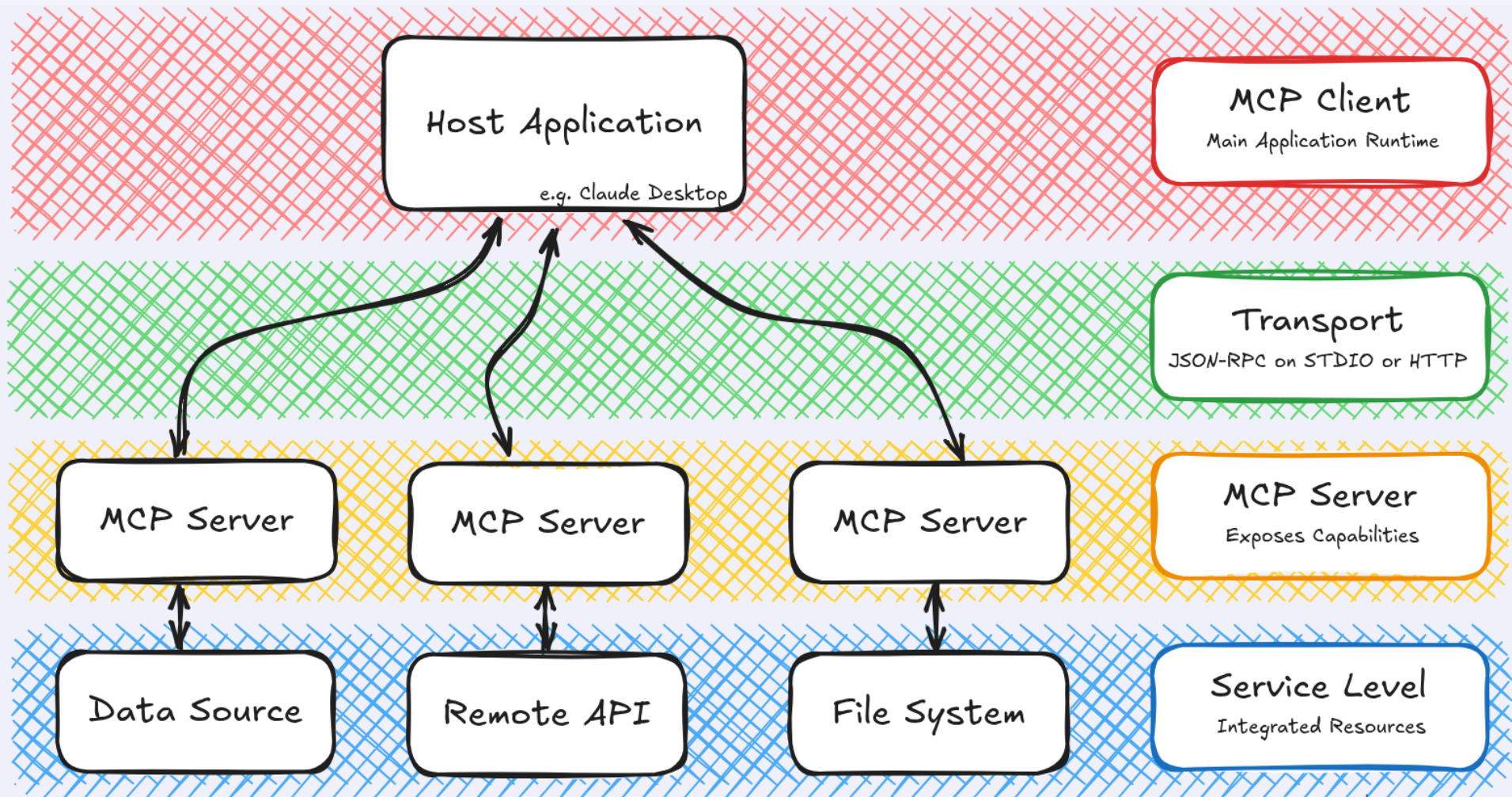


SDK Usage - Server Side

```
use Mcp\Capability\Attribute\McpTool;  
use Symfony\Component\Clock\ClockInterface;  
  
#[McpTool('clock', 'Returns the current date and time.')]  
final readonly class Clock  
{  
    public function __construct(private ClockInterface $clock)  
    {  
    }  
  
    public function __invoke(): string  
    {  
        return $this->clock->now()->format('Y-m-d H:i:s');  
    }  
}
```

Wait?! What's the difference?

General Architecture



Thank You!

Any Questions?

Useful Links

- *Symfony AI*: github.com/symfony/ai
- *Other Libs*: github.com/php-llm/ecosystem
- *Model Context Protocol*:
modelcontextprotocol.io

