Compound Agentic AI Applications: From Development to Systems Implications

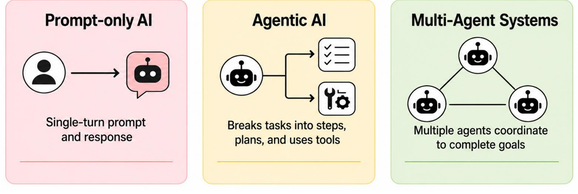
Abdulrahman Alshahrani, under the supervision of Alessandro Cornacchia and Marco Canini

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

SANDS lab

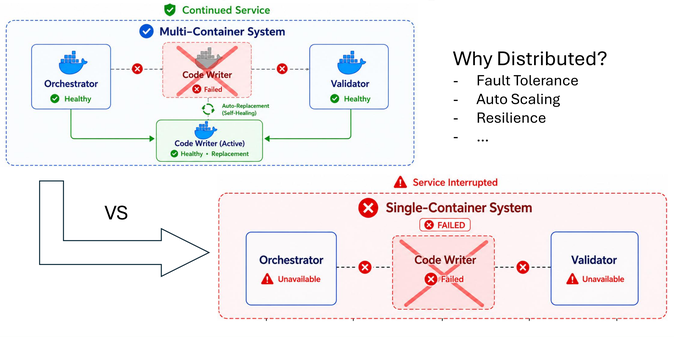
Motivation

Multi-Agent Systems are on the rise



Why? They help decompose hard tasks into simpler problems...

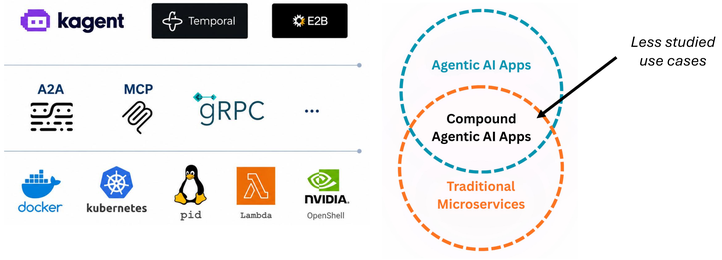
Distributed Multi-Agent Systems are the next stage



Why Distributed?
- Fault Tolerance
- Auto Scaling
- Resilience
- ...

Problem

Developing distributed MAS is non-trivial and rapidly changing



Existing frameworks are constantly updating to support new protocols and runtime environments e.g. LangChain, AutoGen, LlamaIndex

Our research goal:

Can we enable researchers to profile the same agentic applications across different deployment configurations?

DMAS-Forge

Distributed + Multi-Agent Systems = DMAS

"Write once, deploy everywhere"



1. Define workflows

```
func (a *WeatherAgent) Query(ctx context.Context, query string)
(string, error) {
    weatherReport, err := a.llm.LLMCall(ctx, query)
    riskReport, err := a.disasterAgent.Query(ctx, weatherReport)
    return weatherReport + "\n" + riskReport, nil
}
```

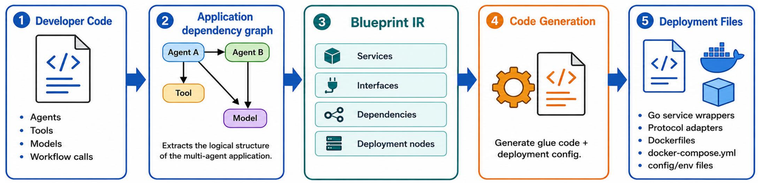
2. Choose how to wire them

```
http.Deploy(spec, weather)
    // a2a.Deploy(spec, weather)
    // mcp.Deploy(spec, weather)

http.Deploy(spec, risk)
    // a2a.Deploy(spec, risk)
    // mcp.Deploy(spec, risk)
```

One-line protocol change

DMAS-Forge Architecture



Features



Example Applications



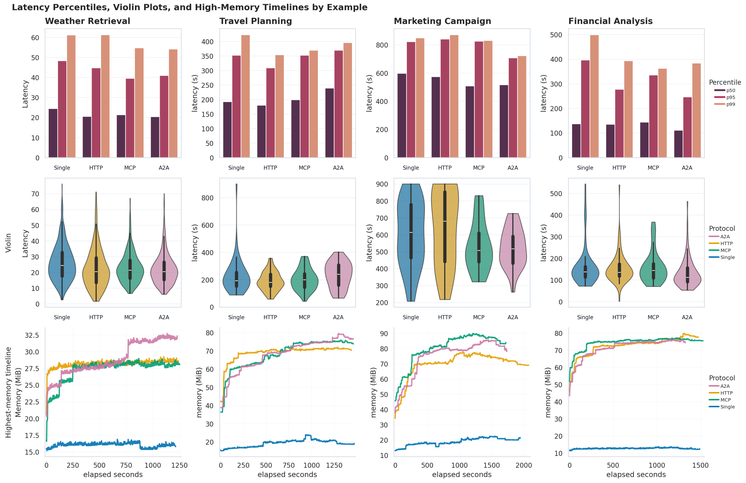
Experiment

Workload: 8 concurrent requests, closed-loop

Timeout: 900 seconds per request

Duration: 20 minutes per deployment type

Model: gpt-oss-120b through OpenRouter.



Total requests: 2227 requests

Failures: 0.7%

Applications	Agent	Operation	Error category	Span Errors
Financial Analysis	Collector	llm.call	Invalid JSON	12
Weather Retrieval	Operator	llm.call	Invalid JSON	7
Financial Analysis	Collector	llm.tool_call	Invalid JSON	5
Financial Analysis	Collector	mcp.tool_call	Invalid JSON	5

Interpretations:

- Single containers have lower throughput and higher tail latencies
- Multi containers consume much more memory vs single-container deployments.
- Multi container protocols offer mixed results between the different applications.

Future works:

- Test different workload patterns (open-loop, Poisson, longer periods)
- Profile "Compound applications" combining agentic and normal workflows.

References

- Cornacchia, A., Anand, V., Bilal, M., Qazi, Z., & Canini, M. (2025). DMAS-Forge: A Framework for Transparent Deployment of AI Applications as Distributed Systems. arXiv:2510.11872.
- Cornacchia, A., Anand, V., Bilal, M., Qazi, Z., & Canini, M. (2025). DMAS-Forge.
- Anand, V., Garg, D., Kaufmann, A., & Mace, J. (2023). Blueprint: A Toolchain for Highly-Reconfigurable Microservices. SOSP '23.
- Cornacchia, A., Anand, V., Bilal, M., Qazi, Z. A., & Canini, M. (2025). DMAS-Forge. SAA '25, Seoul, Republic of Korea.