Next Computing Paradigm

Recommendations for the Next Computing Paradigm

Create digital envelopes

Create a **"digital envelope"** for integrating "anything" (person, company, physical entity, computing device) in a digital space allowing access to multiplicity of services – i.e. building on the concept of "anything-as-a-service" (XaaS) – in an interoperable way. This digital envelope would allow live migration of compute components and a **runtime evolving infrastructure**to support **deployment on a continuum**ranging from resource-constrained edge devices to data centres, allowing for dynamic resource pooling and efficient sandboxed execution of collaborative, migratory compute components offering services.

1. An intelligent **digital agent** able to pursue goals legitimately assigned to it. The intelligent digital agent would be capable of direct execution as well as of **orchestration**. The former would be required when seeking the set goal required local actuation, the latter when the task resulting from the set goal required remote execution.
2. **Sensors**, to pull digitalized inputs from designated sources (in the physical world or other digital envelopes).
3. **Actuators**, to push computed outputs into designated targets (physical things or digital envelopes). The fabric resulting from interconnecting digital envelopes that provide and require services from one another to pursue assigned goals will operate in genuine XaaS modality.

> "Digital enveloping" is the technology-enabled phenomenon by which any item of reality – human, material or immaterial – can be associated with a computable digital representation capable of delegated autonomous action. The notion of delegated autonomous action entails two fundamental traits: that of delegation, which suggests a higher (human) authority that requires some action to be taken (in part) in the digital space; and that of autonomy, which suggests that the pursuit and execution of the required action is carried out by autonomous executable agents that operate within the remits of delegated authority.
>
> The capacity for delegated autonomous action is provided to individual digital envelopes by the combined operation of three key components:

These solutions enable the live migration of compute components across the edge-to-cloud continuum – therefore services – ensuring continuity while addressing latency, privacy, security, risk management, validation mechanisms and context requirements. This is essential to optimize user and infrastructure needs dynamically. In relation to real-world services or actions triggered by a digital envelope, location and time identifiers are assigned, and inter-envelope mechanisms support local vs global optimizations including for safe interactions, managing complex interdependencies and conflict resolution. The next

computing (NCP) exemplifies the notion of continuum. Agreed standards are key for interoperability.

AI-powered orchestrators

Artificial intelligence (AI)-powered orchestrators will be an essential capability of the intelligent digital agent of the digital envelope. AI-powered orchestrators will be developed for the edge – which is strategic as it is located the nearest to the final user – in a manner that can dynamically combine collaborative compute components into executable applications tailored around specific user needs. The task of the orchestrators is to decompose goals set by the user (in the broader term, including human user, company or another permissioned orchestration) into a set of services that cooperate to achieve the set goals. These orchestrators could be themselves generated by (federated) **generative AI (genAI) engines** (supported by more classical algorithmic approaches) located at the edge and capable of collaboration with other orchestrators within federated zones.

Space- and time-aware protocols

Expand and adapt web-level protocols and associated standards by enhancing the existing suite of HTTP-based protocols to be both spatially aware and time-sensitive. This will allow web-level interactions between NCP's migratory compute components to account for 3D physical space and real-time communication, drawing on technologies like WebRTC to manage time-sensitive tasks effectively and the spatial web (IEEE P2874, OpenUSD, …).

Interoperable contract-based API specifications

Establish interoperable, contract-based application programming interface (API) specifications – usable by expanded web-level protocols – ensuring that interconnected services communicate with clear expectations of both functional and non-functional performance. These contracts, similar to service-level agreements (SLAs), should detail the conditions under which services will optimally perform, including non-functional requirements, ensuring smooth integration and reliable service delivery within the NCP framework. These APIs should account for non-functional properties like latency, cost, and performance. The resulting model should ensure that an API not only promises to deliver a service but also specifies the conditions under which it can perform optimally. The API should also be compliant with the currently proposed API for large language models LLMs `[OpenAIFunction] [BerkeleyFunction]`.

Promoting these standards in relevant standardization bodies is essential for fostering interoperability and consolidating development conditions through standardized benchmarks, testing methodologies, and best practices. This will ensure that implementations can be effectively and securely integrated, improving overall system efficiency and reliability, and enabling the creation of an interoperable business ecosystem of services and orchestrators.

Introduction

NIST Recommendation SP 800-145 [NIST], dated 2011, lists five defining traits that characterize cloud computing:

1. on-demand self-service,
2. broad network access,
3. resource pooling,
4. rapid elasticity,
5. measured service.

Back then, features (1), (3) and (4) were by far the most visionary ones in terms of (provider-side) requirements and (user-side) expectations. In fact, their pursuit has had a major impact, shaping a whole new world of cloud-enabled technology in the subsequent decade.

Feature (1) implies that, rather than (application) services having to be installed, they would be delivered via the web, that is, via client-side web browsers that consequently became "versatile self-contained fully-provisioned application environments". All the client side needs in the cloud model is a cloud-enabled web browser and broad network access. This notion has had vast consequences and stands at the basis of the NCP vision, as discussed in the following section.

Features (3) and (4), largely immaterial to the client side, concern primarily what the provider platform must be able to do. Resource pooling is the principle by which the provisioning and apportionment of computing, storage and networking is no longer confined to a single physical place. In the cloud model they become virtual units, which result from concrete fragments opportunistically scattered in multiple places, located wherever there is a convenient temporary "home" for them.

> The term "home" is meant here to designate infrastructure resources (compute, storage, and networking) able to support the deployment and the execution of the digital entities that are being pooled. An analogy may clarify. In an operating-system environment, memory is made available to executing processes as a logical resource that virtualizes physical memory. No

single process actually owns physical memory, which is divided in page frames handled by the OS. Processes are loaned (sparse) page frames, strictly on the base of need, to host page contents coming from and going to secondary storage. Resource pooling in the cloud essentially follows the same concept, except spanning over networked nodes, which effectively means virtualization over the network, beyond the physical boundaries of a single computer.

Earlier computing models had already long known and practised virtualization, which has since become the foundation of the computing stack in the guise of virtual memory, pre-emptive scheduling, file systems, to name just a few.

The dominant interpretation of the cloud as a concrete provisioning platform soon became that of the giant corporations that saw and developed the web as their marketplace. In that view, resource pooling would be achieved by amassing and virtualizing immense clusters of comparatively cheap networked computers deployed at strategic locations. At that point, computing would happen on any of such clusters (at the notional centre of the cloud) and data would flow there from its sources (at the notional edge of the network). Users at the client side would only need web browsers on their devices to be able to use rich, reactive, sophisticated single-page web applications, while most of the juicy action would happen on the server side at the centre of the cloud.

The cited NIST Definition also posits that cloud computing has three service models: software-as-a-service (SaaS); platform-as-a-service (PaaS); infrastructure-as-a-service (IaaS). The SaaS model was the most obvious and immediate one to be understood, as it speaks directly to the end user. The IaaS model allowed enterprises to conceive and deliver SaaS offerings without owning concrete infrastructures and yet being able to control rental costs. In fact, it was the IaaS model and not the SaaS that allowed thriving digital businesses to emerge.

It is now technically possible and strategically opportune to separate what is specific to the defining traits of the cloud in the “traditional” model from what can be realized in alternative modalities. Doing that opens up novel and unprecedented opportunities that belong in the HiPEAC Vision.

The traditional model places at the centre of the cloud the centre of gravity of computing. The user and the data are attracted to gravitate towards and around it. That tenet carries the view that the “important” computing resources are available solely at the centre of the cloud. That is the fundamental premise to monopoly, which is what we have observed in the cloud offering in the last decade.

The fact is however that at present a vast cumulative amount of computing resources is available at the edge of the network, where users and data sources are.

Consider the total number of computers embedded in cellular phones (7.2 billion in use worldwide in the year 2024), modern transportation vehicles (several hundreds of million times ten or more per vehicle) both mobile and stationary, home automation systems. Imagine some of their resources pooled together, opportunistically around geographically close zones, to host edge-related applications. The infrastructure resulting from this virtual pool would never compete with cloud-enabled data centres, purpose-built to support enterprise-level applications and large-scale data processing. And never it should, in fact, as edge-friendly applications are nimble and low latency, which is quite the opposite extreme to them.

If those resources were pooled together seamlessly, à lacloud, innumerable value-added computations could take place at the edge instead of at the centre of the cloud; see Figure. That shift would prize privacy, latency, energy, decentralization, personalization, context-awareness in a manner that the centre of the cloud could not possibly match.

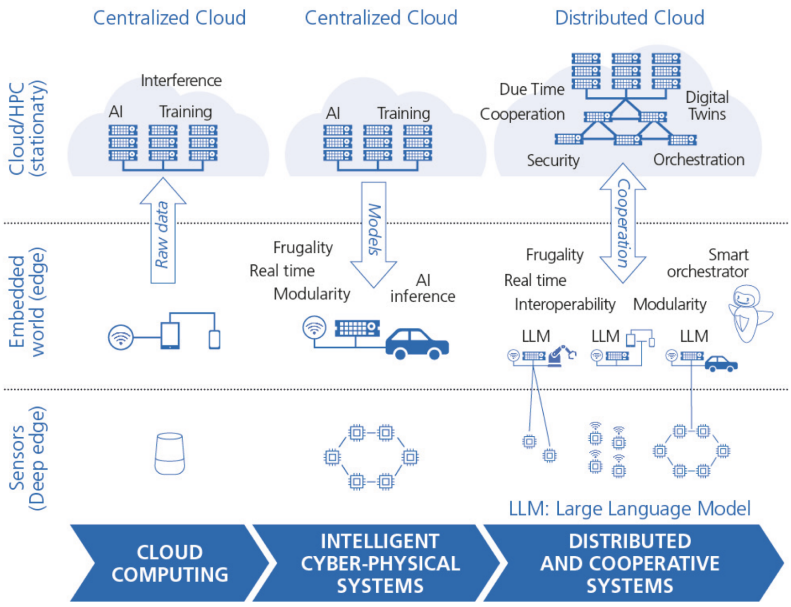

Figure 1: Evolution of computing infrastructures towards the NCP, where services are distributed and cooperate together. Credit: Denis Dutoit, CEA

The established principles of resource pooling and virtualization applied to the edge would allow the creation of malleable, powerful, dynamic, virtually ubiquitous federations of edge nodes strategically positioned where the physical world borders the digital sphere. This connotation is essential to the NCP as digital resources capable of sensing and actuation, in addition to computing, may interact with entities in the physical space causing the physical and the digital worlds to come together seamlessly and dynamically.

Pooling edge resources among themselves and with the cloud seamlessly gives rise to the so-called edge-cloud continuum, a compute infrastructure where computation would be deployed opportunistically and dynamically, wherever that is more convenient for the user.

Digital envelope

“Digital enveloping” is the technology-enabled phenomenon by which any item of reality, human, material and immaterial, can be associated with a computable digital representation capable of delegated autonomous action.

> The notion of delegated autonomous action entails two fundamental traits: that of delegation, which suggests a higher (human) authority that requires some action to be taken (in part) in the digital space; and that of autonomy, which suggests that the pursuit and execution of the required action is carried out by autonomous executable agents that operate within the remits of delegated authority.

The capacity for delegated autonomous action is provided to individual digital envelopes by the combined operation of three key components; see Figure:

- An intelligent **digital agent** able to pursue goals legitimately assigned to it. The intelligent digital agent would be capable of direct execution as well as of **orchestration**. The former would be required when seeking the set goal would require

local actuation. The latter when the task resulting from the set goal would require remote execution.

- **Sensors**, to pull digitalized inputs from designated sources (in the physical world or other digital envelopes).
- **Actuators**, to push computed outputs into designated targets (physical things or digital envelopes).

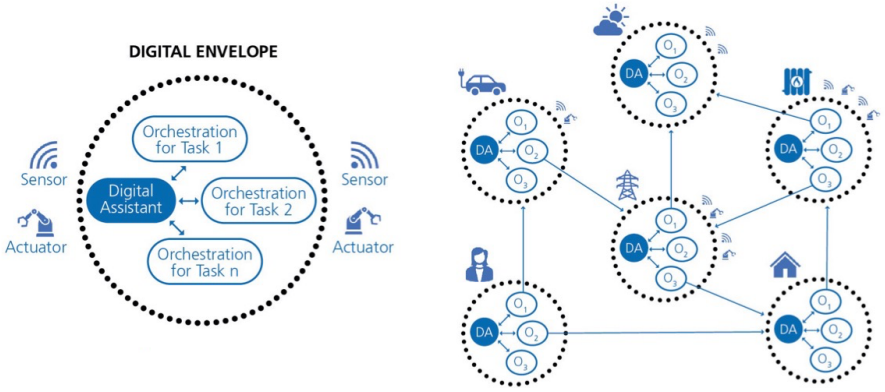

Figure 2: Digital envelopes interacting together

The fabric resulting from interconnecting digital envelopes that provide and require services from one another to pursue assigned goals will operate in genuine XaaS `[XaaS]` modality.

The web has shown that digital resources can be given uniform representations and identities, and can be operated upon by CRUD (create-read-update-delete) service primitives exposed by way of HTTP verbs `[Kann]`. Digital envelopes would thus be woven into a next-generation web `[HV23NextWeb]`, which brings together the web of humans with the digital web, into a programmable and interoperable hyperspace. The XaaS paradigm emanates from that notion as a major vector of innovation, which shifts the centre of gravity away from the cloud towards the edge.

The compute component of digital envelopes must instead be capable of live migration on a **continuum runtime infrastructure** spanning from resource-constrained edge devices to data centres. That capability is a direct consequence of allowing for orchestrated actions to be deployed wherever their execution is best assigned, where the notion of “best” may even change over time.

The continuum infrastructure should allow for dynamic resource pooling and efficient sandboxed execution of collaborative, migratory service-providing and service-requiring compute components. Live migration of compute components across the edge-to-cloud continuum, therefore services, will ensure continuity of service while addressing latency, privacy, security, risk management, validation mechanisms and context requirements. This capability is essential to optimize user and infrastructure needs dynamically.

Digital envelopes would have owners, who should be the sole entity authorized to communicate goals to them. The digital agent of the digital envelope should receive those goals and translate them into a permissioned orchestrations of request-response interactions with other digital envelopes (thereby with the digital agents within them). Thanks to actuators, those interactions may take effect on the physical world or on the digital sphere or both. Those effects might be “sensed” by other digital envelopes and possibly further “acted” upon to adjust to emerging needs arising as a function of local and global constraints.

Digital envelopes evolve the concept of “digital twin” in scope and capability. In scope, no longer confined to an encapsulated digital sphere, but capable of actuation into the physical world. In capability, via the capability of autonomous planning and execution in pursuit and accomplishment of assigned goals.

A simple example might help illustrate the concept of the digital envelope.

Use case: Travel

Travelling independently for disabled people (wheelchair users, visually impaired people, …) is a challenge. The personal agent, operating from within the digital envelope, and the NCP, can help such people to travel independently, as the following scenario illustrates. Before leaving, the personal agent will produce a travel plan, based on the starting point and the destination, including all the assistance needed during the trip. While travelling, the personal agent will continuously update the travel plan based on actual information.

In a typical travel scenario, the personal agent will instruct the orchestrator to call a taxi to drive to the station. It will make sure that (i) the taxi has the space to take a wheelchair on board, and (ii) the taxi arrives at the station on time. Before arriving at the station, the personal agent contacts the digital envelope of the train station and arranges assistance to get to the right platform and to board the train. On the train, the personal agent contacts the digital envelope of the train, which in turn contacts the digital envelope of the staff of the train. In case the traveller needs assistance, they can ask their agent to contact the train staff. At the destination, the personal agent will again contact the digital envelope of the train station and order local assistance. If special assistance is not required, the agent will help the traveller find the best route to their destination (wheelchair accessible, adapted to blind travellers, …).

On arrival at the destination, the personal agent will look for a place to eat. It will only show the restaurants that are wheelchair accessible, and that offer items that are compliant with the dietary requirements and the preferences of the traveller.

The personal agent will also take care of all the tickets and payments. This means that travellers can freely use any bus, tram, metro, shared bike, or enter a museum without having to worry about the payment. During ticket inspection, the inspector or the inspection machine will directly talk to the digital envelope of the traveller.

When renting a car, the personal agent will take care of all the "paperwork" ahead of time, and there is no longer the need to pick up the keys. The personal agent will directly talk to the digital envelope of the car and give the driver access to the car. The personal assistant will also help the driver to operate the car by answering questions, or, in some cases, by taking actions ("I will switch on the fog lights for you"), or by warning the driver ("it is better to recharge here because the next charging station is 200 km away"). In a future scenario the personal agent will instruct the digital envelope of the car to autonomously drive to the destination.

Among other things, the use-case scenarios of the digital envelope discussed in this document show that a large fraction of the (compute-and-communicate) actions pertinent to achieving a user-related goal ought to occur near or at the edge. They further posit that certain edge nodes may need to be able to aggregate opportunistically into ephemeral (temporary) federations to accomplish assigned goals in a manner that respects legal and physical boundaries and constraints, and that seeks some definition of overall efficiency.

There is very clear correspondence between the vision outlined above and the fast-rising momentum of “agentic AI”.

For an explanation of the notion of “agentic AI”, see for instance [ErikPounds], although note that this piece – owing to the identify of its editor – suffers some commercial bias.

The APA Dictionary of Psychology defines “agentic” as a psychological condition that occurs when individuals, as subordinates to a higher authority in an organized status hierarchy, feel compelled to obey the orders issued by that authority [APA]. When used in the AI context, the “agentic” term thus is loaded because psychology associates it with potentially negative connotations (destructive obedience), which suggests extreme caution when deployed in digital programs that are bound to act much faster and deeper than human mind can comprehend.

Agentic AI is very clearly the next frontier of genAI, moving it beyond the request-response modality that it has had so far, which has been shown to lack scalability, and to be confined to cute but limited code assist, customer service, and content writing service contexts. The current genAI model is that of pipeline where:

1. a request is initiated via a natural-language, written or oral, prompt;
2. relevant data is accessed through a retrieval-augmented generation, RAG;
3. an answer is returned, which may be right (accurate, pertinent) or wrong (inaccurate, not pertinent, erroneous).

The new model of agentic AI uses genAI to draw and execute a plan to perform work that is to meet user-specified goals. In doing so, the digital entity (which the agentic AI literature calls “agent”, causing the reader to believe that “agent” is synonym to “agentic”, which it really is not) may work in concert with other such digital entities as part of an orchestration of interactions expected to deliver coordinated outcomes. It should be noted that this notion of orchestration has been the focus of attention of several prior editions of the HiPEAC Vision [HV21Angels], [HV23Digels]. It should also be noted that orchestrations can be realized as hierarchical descents, from a higher-level (more abstract) set of goals into lower-level (increasingly more concrete) set of either simple tasks or even other orchestrations. Some such orchestrations may coalesce into advertised capabilities, as permanent entries into public registries.

Early demonstrators have been released lately by various actors at the forefront of genAI, which show glimpses of what the evoked scenarios may give rise to; see for example: [Gorilla], [Magnetic-One]. Interestingly, the most profound implication of these developments is that the entire technology stack needed for all these digital envelopes to be deployed and executed (prompts, routines, tools, function schemas, handoffs, etc.) might be generated on the fly ad infinitum, as part of the mechanics of turning goals into plans, and acting in response to contingencies resulting from actions along the pipeline.

To sum up

The potentially cascading or federated orchestration discussed in this chapter will have to keep an efficient balance between resource availability (which values the centre of the cloud), and privacy, latency, energy, decentralization, personalization, context-awareness (which prizes the edge). That will have to be much more dynamic and adaptive than traditional orchestration at the centre of the cloud. The resulting orchestrations would be dynamic, opportunistic, ephemeral, and maximally loosely coupled, in addition to collaborative (and thus hierarchical or federative or both). The associated computations (tasks) should be able to move across the continuum in search of the temporary residence best fit to meet stated goals.

The envisioned orchestration would embed intelligence, including next-generation genAI, to do the bidding of individual users at the edge, prompted by user goals and requirements and

returning ad hoc programmatic orchestration engines. The underlying infrastructures would also need intelligence to federate opportunely and adaptively available resources.

This model entails a whole a new frontier for computation, computing artifacts, and computing infrastructure, which this document calls the next computing paradigm (NCP). Realizing the NCP requires evolving the runtime infrastructures availed at the edge. It also requires expanding the web-level suite of protocols in order to become spatially aware, so that the web becomes a 4D place (aware of the three dimensions of our physical reality, plus time-sensitive) that all interactions, node-to-node, client-server, machine-to-machine may be carried by HTTPs-based bidirectional multiplexed server-prompted and asynchronous channels.

Conclusion

In order for the vision outlined in this chapter to be brought to fruition, certain specific routes of innovation would have to be taken. We list them next in no particular order.

Runtime infrastructures fit for deployment onto resource-scarce compute devices at the edge should be developed, making them capable of supporting dynamic resource pooling and of hosting efficient sandboxed execution of migratory collaborative compute components.

> Migration is an essential trait of the opportunistic federation of computing entailed by the notion of digital envelopes: actions must be taken at places that depend on the set goals and on the logistical constraints (in the physical or digital world or both). An orchestrated action must therefore be dispatched for execution at a place that is other from that of residence of the orchestrator.

Solutions that allow compute components to live migrate across the edge-to-cloud continuum should be developed that warrant continuity of execution, whenever transfer to a different node may warrant superior coverage of user- and infrastructure requirements such as latency, privacy, security, provenance, context, etc.

The web-level suite of HTTP-based protocols should be expanded and streamlined to make them (1) spatially aware, so that web-level interaction between migratory compute components is aware of 3D physical space, and well as (2) time-sensitive, learning from the real-time capabilities of e.g., WebRTC.

API description standards such as `[OpenAIAPI]` are currently being made obsolete by “function schemas” or equivalent technicalities that revolve on a local and opportunistic need to describe APIs to LLMs so that the latter can incorporate the former into responses to prompts, and generate actions from goals, which call them at the appropriate place. What is needed, instead of local, ad hoc, half-baked solutions, is a concerted design effect that determines how best to describe APIs so that all requirements discussed in this document can be met satisfactorily and in an open, interoperable manner.

The API description standard of interconnected web-level services should be augmented with interoperable contract-based specifications, akin to service-level agreements (SLAs) or assume-guarantee pairs, to ensure that required and provided services can communicate with an expected level of functional and non-functional performance. The resulting model should ensure that an API not only promises to deliver a service but also specifies the conditions under which it can perform optimally. This requirement of functional and non-functional interoperability extends to the modality of interconnection between LLMs, which currently goes under the name of “function calling”.

Solutions that allow the embedding of genAI at the edge should be developed in order that human users can be provided with natural interfaces (voice, gesture, eye movements, touch) to the digital world, with more energy efficiency, reduced latency, lesser communication overhead, and greater privacy.

AI-powered edge-based orchestrators should be developed that reflect the vision discussed in this chapter. These should be capable of dynamically combining migratable collaborative compute components into ephemeral, opportunistic smart personalized applications in response to user requirements. Those orchestrators should be the programmatic output of genAI engines located at the edge and should be able to collaborative with other such engines located within federative zones.

Demonstrable proof-of-concept implementations should be developed based on elements of the capabilities evoked in this chapter, whether limited to selected features only or on a more holistic scale, in articulations that are not proprietary and support open standards and platforms.

References

`APA:` American Psychological Association. APA Dictionary of Psychology: “agentic stage”. https://dictionary.apa.org/agentic-state

`BerkeleyFunction:` Berkeley Function-Calling Leaderboard. https://gorilla.cs.berkeley.edu/leaderboard.html

`ErikPounds:` Erik Pounds @ NVIDIA. “What Is Agentic AI?”. October 22, 2024. https://blogs.nvidia.com/blog/what-is-agentic-ai/

`Gorilla:` Shishir G. Patil and Tianjun Zhang and Xin Wang and Joseph E. Gonzalez: “Gorilla: Large Language Model Connected with Massive APIs”. arXiv, May 24, 2023. https://arxiv.org/pdf/2305.15334

`HV21Angels:` Marc Duranton and Tullio Vardanega: “Guardian Angels” to protect and orchestrate cyber life. HiPEAC Vision 2021 (Pages 50-55). https://www.hipeac.net/vision/2021.pdf

`HV23Digels:` Tullio Vardanega and Marc Duranton: “Digels”, digital genius loci engines to guide and protect users in the “next web”. HiPEAC Vision 2023 (Pages 18-21). https://www.hipeac.net/vision/2023.pdf

`HV23NextWeb:` HiPEAC Vision 2023. The Race for the “Next Web” (pages 13-64). https://www.hipeac.net/vision/2023.pdf

`Kann:` Charles W. Kann III: §6.1 CRUD Interface. LibreTexts Engineering. https://shorturl.at/cb0Wp

`Magnetic-One:` Adam Fourney and Gagan Bansal and Hussein Mozannar and Victor Dibia and Saleema Amershi: “Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks”. Microsoft. AI Frontiers blog, November 12, 2024. https://shorturl.at/ihBLx

`NIST:` NIST Special Publication 800-145: The NIST Definition of Cloud Computing. September 2011. https://doi.org/10.6028/NIST.SP.800-145.

`OpenAIAPI:` https://www.openapis.org/

`OpenAIFunction:` OpenAI Platform: Function Calling. https://platform.openai.com/docs/guides/function-calling

`XaaS:` Short for “Anything-as-a-Service”. See for example: https://www.digitalroute.com/resources/glossary/xaas/