____2024

European Patent Office (Munich)
80298 Munich
Germany

European Patent Application No.: 19830491.7
Applicant(s): Krishnan, Anantha Chathanur Raman

Dear Sirs,

In reply to the Communication Pursuant to Article 94(3) EPC dated July 04, 2023, issued by the European Patent Office, Applicant hereby submits response to objections/observations and requests our response to be taken on record for further prosecution/examination of the application.

We herewith filed an amended set of claims 1-13 that are to replace claims 1-13 currently on file.

AMENDMENTS TO CLAIMS AND BASIS THEREOF

1. Independent claims 1 and 11 have been amended to further clarify the claimed subject matter. Support for the amendments can be found at least in previously presented claim 2; and from the paragraphs [00027], [00077], of the as-filed specification.
2. Dependent claims 2, 3, 8 have been amended to clarify the claimed subject matter and to remove defects in form or content of claimed subject matter.
3. As can be seen, no new matter has been added to the claims.

Item 2 and 3- Lack of Clarity and Conciseness (Article 84 EPC)

The substantive examination report dated July 04, 2023, contains the following observations and objections:

Sections 3, 3.1, 3.1.1 and 3.1.2 of the search report assert that- "The application does not meet the requirements of Article 84 EPC, because claims 1-13 (of the corresponding EP Application no. 19830491.7) are not clear:"

Sections 3.1.1 of the substantive examination report asserts that the terms "interest list", an "SR belong[ing] to an asset class" and "attribute-based interest configuration" are vague and unclear (Article 84 EPC).

Applicant respectfully disagrees and submits that the published application at multiple instances provides sufficient description to clearly define the scope and meaning of the terms above.

Paragraph [00034] of the as-filed specification states that-

> "at least one node of the one or more available nodes can be owned by a node owner, the node owner being able to configure **an interest list** for the at least one node such that the at least one node can process a given SR based at least on the interest list in a manner that if the given SR belongs to an asset class that forms part of the interest list, the at least one node can process the given SR."

Paragraph [000105] of the as-filed specification states that-

> "Figure 7 illustrates the management of existing and onboarding nodes in the network. A node owner 702 acquires a hardware 704 and may install the relevant software supporting either service, compute or storage layer. Once the hardware 704 is provisioned, the node owner 702 may **specify any number of transaction types of assets classes that they are interested in the node to process, as shown at interest list 706**. Once the interest is established, the node is made to go through a benchmark test to ensure that it is capable of servicing all the claimed interests, as shown at 708."

Further, paragraph [000134] of the as-filed specification states that-

> "It offers a multi-dimensional value environment for all actors in the network to leverage their respective interests and abilities to leverage services, in a just manner without the need for third party rating agencies."

Further, paragraph [000140] of the as-filed specification states that-

> "Proposed system can attribute a node's runtime behaviours such as, but not limited to, the **economic and asset type based interest** by vesting Native Virtual Currencies and/or heterogenous tokens, sincerity, reliability, availability etc. into an alphanumeric value called to the node as its Node Trust Quotient (NTQ) used to reflect the node's runtime behaviours."

At least based on the above, and as shown in FIG. 7 ("interest list" 702) refers to a configurable set of preferences created by the owner of at least one network node. This list outlines specific transaction types or asset classes that the node owner is interested in processing. The purpose of the interest list is to enable a node to handle a given Service Request (SR) in a manner aligned with the specified preferences. If the Service Request (SR) pertains to an asset class included in the interest list, the node processes it accordingly. Further, Service Request (SR) belonging to an asset class signifies a Service Request associated with a particular category or type of asset. The interest list serves as a tool for the node owner to articulate preferences regarding the types of transactions or asset classes they want their node to engage with. When a given Service Request (SR) falls within the defined asset classes in the interest list, the node processes the request in accordance with the established criteria, as indicated in the specification.

Thus, the applicant submits that, at least in the above paragraph, the meaning of "interest list" and "Service Request (SR)" are clear and is not vague. Further, Applicant has amended the claim 8 to clarify the technical feature. Further, Applicant had previously presented amended independent claims 1 and 11 to remove "attribute-based" to further clarify the technical feature. In view of the above, Applicant respectfully submits that the claimed feature is clear and not vague.

Sections 3.1.2 of the substantive examination report asserts that- "Mutatis mutandis, the clarity objections against claim 1 hold against its dependent claims, as well as against independent claim 11 and its dependent claims which thus do not meet the requirements of Article 84 EPC."

In response, Applicant had previously presented amended independent claims 1 and 11 to remove "attribute-based" to further clarify the technical feature and to remove all instances of the

acronyms for clarity purposes. In view of the above, Applicant respectfully submits that the claimed feature now is clear and not vague.

Claim 3

Sections 3.2 of the substantive examination report asserts that the examining division maintains the objection raised in the European Search Opinion of 24.02.2023 (ESOP), sections 3.3, 3.4 and 3.10.

Sections 3.3 of the (ESOP) asserts that- "The terms 'interaction vector', 'cause/intent behind the SR' and 'depiction of why the SR was executed' are vague and unclear and leave the reader in doubt as to the meaning of the technical feature to which they refer, thereby rendering the definition of the subject-matter of said claim unclear (Article 84 EPC). In particular, it remains unclear what contents and origin of the 'interaction vector' are and how 'cause/intent behind the SR' and 'depiction of why the SR was executed' are expressed by way of technical features."

Paragraph [000112] of as-filed specification states that-

> "Figure 9 illustrates how super state structures are created. The client requests/transactions not only carry payload data but also metadata as shown in the note. These metadatas are collectively processed in parallel to formal verification and finally classified into a data structure called "Interaction Vectors", as shown at 902. The Interaction vectors are the data structures applied on to the raw transactional payload to derive full context of the transaction such as Identity vectors (904), Causal vectors (906), and Method vectors (908). The attributes of the interaction vectors pointing to the user wallet can be updated by application of new information sourced by the same vectors."

Further, paragraph [000129] of the as-filed specification states that-

> "In another exemplary embodiment, the proposed system can include special algorithms represented as "Interaction Vectors". The vectors can be applied on transactions propagated by the actors involved in a peer to peer cryptographic network connecting devices with exclusive computational capabilities that can record transactions as and when occurred. The connected devices can collectively execute and apply the vectors onto the receiving transactions that are signed by the actor's account credentials. The harvested values from the application of the interaction vectors can be attributed to the actor's account responsible for the same transaction. The values can be updated in an array of key-value pairs, with each key representing a disparate inter-dimensional value. The transaction

> descriptors can be updated upon new entry to the array or update of value to an existing element with the same key. The disparate values in the descriptors can be transferred either at a mutual exchange rate to the native virtual currency of the network, or any of the heterogenous tokens."

Further, paragraph [000136] of the as-filed specification states that-

> The technical advantage of the proposed system lies application of interaction vectors on the heterogenous transactional payload of various encoding types in the form of data harvesting and pattern matching algorithms. The system has the ability to recognize the actors responsible for the transactions—the 'from' accounts propagating the transaction and the 'to' accounts being the beneficiary of the propagated transaction, through the application of "Identity Vectors". Further, the system has the ability to capture the intent behind a propagated transaction, through the application of "Causal Vectors", and the ability to capture the means to achieve the fulfillment and finality of the transactions, through the application of "Method Vectors".

In view of at least the above cited paragraphs and Figure 9 of the as-filed specification, Applicant respectfully submits that interaction vector has been defined in as-filed specification as "…the data structures applied on to the raw transactional payload to derive full context of the transaction such as Identity vectors (904), Causal vectors (906), and Method vectors (908)."

The system utilizes "Interaction Vectors" to process transactions in a peer-to-peer cryptographic network. These vectors consist of metadata associated with client requests or transactions, processed alongside formal verification. The outcome is a set of data structures, namely "Interaction Vectors," which include Identity Vectors, Causal Vectors, and Method Vectors. Identity Vectors help identify the actors involved in transactions, specifying the 'from' and 'to' accounts responsible for initiating and receiving the transactions, respectively. Causal Vectors capture the intent behind a transaction, providing insights into why it occurred. Finally, Method Vectors document the means employed to execute and complete the transactions.

Accordingly, Applicant respectfully submits that the definition of the claimed subject-matter is clear as to the contents and the way the technical features are expressed in the claim.

Claim 4

Sections 3.4 of the of the (ESOP) asserts that- "The terms 'economic value', 'right of possession', 'emotion parameters', 'social parameters', 'time value' and 'privilege dimension' are vague and unclear and leave the reader in doubt as to the meaning of the technical feature to

which they refer, thereby rendering the definition of the subject-matter of said claim unclear (Article 84 EPC)."

Paragraph [000114] of the as-filed specification states that-

> "Figure 10 illustrates some of the attributes of the super state structures (SSS). The super state structure can host more than one measurable attributes that can be used by the user while creating a transaction or client service request. All the attributes of the super state structure related to users are persisted in the respective user's wallet, whereas the super state structure related to the entire system is persisted directly in the distributed storage network to support operations carried out by the Distributed Compute network. As illustrated, attributes of SSS can include Economic value, Privilege Dimension, Right of possession, Time value, Emotions and Social, amongst others."

Further, paragraph [000112] of the as-filed specification states that-

> "Figure 9 illustrates how super state structures are created. The client requests/transactions not only carry payload data but also metadata as shown in the note. These metadatas are collectively processed in parallel to formal verification and finally classified into a data structure called "Interaction Vectors", as shown at 902. The Interaction vectors are the data structures applied on to the raw transactional payload to derive full context of the transaction such as Identity vectors (904), Causal vectors (906), and Method vectors (908). The attributes of the interaction vectors pointing to the user wallet can be updated by application of new information sourced by the same vectors."

In view of at least the above cited paragraphs and Figures 9 and 10, the terms "economic value", "right of possession", "emotion parameters", "social parameters", "time value" and "privilege dimension" are treated as attributes within "interaction vectors." These vectors are data structures applied to the raw transactional payload to understand the full context of the transaction. So, the terms "economic value", "right of possession", "emotion parameters", "social parameters", "time value" and "privilege dimension" are elements considered and represented within the "interaction vectors" for a comprehensive understanding of a transaction or client service request.

Accordingly, Applicant submits that meaning of the technical feature to which the above terms refer define the subject-matter clearly.

Items 4-6: Novelty and Inventive Step as per European Search report

D5- Anonymous: "Publish-subscribe pattern - Wikipedia",
2 December 2015 (2015-12-02), pages 1-6, XP055879072, Retrieved from the Internet: URL: https://en.wikipedia.org/w/index.php?title=Publish-subscribe_pattern&oldid=693442115 [retrieved on 2022-01-14]

Independent claims 1 and 11

Applicant respectfully submits that amended independent claim 1 is novel and inventive over D1-D5 because D1-D5 do not disclose "*a system state manager (302) that, based on at least one attribute of each service request in the interaction pool, identifies a compute node (404) from one or more available compute nodes comprised in the distributed compute (304), that has an ability and an interest list configuration to execute said respective service request, and transmits said respective service request to said identified compute node (404) for execution, wherein said interest list configuration indicates an asset class of service requests that said identified compute node is configured to execute, and wherein said distributed compute (304) is operatively and communicatively coupled with a distributed storage (308), said distributed storage (308) being configured to store a super state structure (306) representing metadata associated with said service request or output of said execution of said service request by said identified compute node (404)", as recited in amended independent claim 1 (emphasis added).*

With respect to previously presented independent claim 1, the substantive examination report dated July 04, 2023, on page 2, concedes that "D1 is silent on determining an available processor node that has any 'interest list' configuration for execution of any task or service request" and that "[a]ccordingly, D1 fails to disclose 'a system state' manager (302) that, based on at least one attribute of each service request in the interaction pool, identifies a compute node (404) from one or more available compute nodes comprised in the distributed compute (304), that has an ability and an interest list configuration to execute said respective service request, and transmits said respective service request to said identified compute node (404) for execution,

wherein said interest list configuration indicates an asset class of service requests that said identified compute node is configured to execute'.

Further, the substantive examination report on page 2 "acknowledges that D1 is silent on any interest list. The difference thus constitutes novelty". The Applicant agrees.

However, the examiner in page 2 and 3 of the substantive examination report asserts that the difference to be non-inventive and asserts that "the examining division acknowledges that the technical effect achieved by the above difference is the one stated by the applicant in LET, namely "tak[ing] into account any interest configuration (i.e., type of transaction or asset class of service requests) of the nodes." However, message schemes, where nodes can express their interest in certain messages resp. service requests, were already known at the time of priority of the application and thus belong to the general knowledge of the person skilled in the art.

The examiner asserts that the document D5 in page 1, lines 7-12 discloses "In software architecture, publish-subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead characterize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers' express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are."].

The Applicant respectfully traverses the above contentions of the examiner and asserts that the subject matter of claims 1 and 11 involves an inventive step within the meaning of Article 56 EPC. The detailed arguments submitted on 23 September 2023, in support of the inventive step of the independent claims, are maintained, mutatis mutandis. In addition, further arguments are presented below.

D5 in page 1, lines 7-12 asserts:

> "In software architecture, publish-subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead characterize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are".

D5 in page 1, lines 13-19 asserts:

> "Pub/sub is a sibling of the message queue paradigm, and is typically one part of a larger message-oriented middleware system. Most messaging systems support both the pub/sub and message queue models in their API, e.g. Java Message Service (JMS). This pattern provides greater network scalability and a more dynamic network topology, with a resulting decreased flexibility to modify the Publisher and its structure of the data published".

D5 in page 2, lines 1-15 asserts:

> "In the pub/sub model, subscribers typically receive only a subset of the total messages published. The process of selecting messages for reception and processing is called filtering. There are two common forms of filtering: topic-based and content-based. In a topic-based system, messages are published to "topics" or named logical channels. Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe, and all subscribers to a topic will receive the same messages. The publisher is responsible for defining the classes of messages to which subscribers can subscribe. In a content-based system, messages are only delivered to a subscriber if the attributes or content of those messages match constraints defined by the subscriber. The subscriber is responsible for classifying the messages. Some systems support a hybrid of the two; publishers post messages to a topic while subscribers register content-based subscriptions to one or more topics".

The applicant submits that the substantive examination report seemingly equates "senders" as disclosed in D5 to "system state manager" as recited in claim 1 and the "receivers" as disclosed in D5 to a "compute node" that has expressed a corresponding interest in "asset class of service requests" recited in claim 1.

D5, in the cited portions and elsewhere, describes publish-subscribe (pub/sub) messaging pattern in software architecture that involves communication between publishers, who send messages without specifying direct receivers, and subscribers, who express interest in specific classes of messages. This pattern operates independently of the knowledge of potential subscribers or publishers. Pub/sub is part of message-oriented middleware systems and coexists with the message queue paradigm.

In addition, D5 further describes that in pub/sub, subscribers typically receive a subset of total published messages, and filtering is the process of selecting messages for reception and

processing. For example, D5 defines two common filtering forms, topic-based and content-based. In topic-based systems, messages are published to named logical channels (topics), and subscribers receive all messages from subscribed topics. On the other hand, content-based systems deliver messages to subscribers based on predefined constraints, with subscribers responsible for classification. Some systems may combine both approaches, allowing publishers to post messages to topics while subscribers register content-based subscriptions to one or more topics.

D5, merely describes a pub/sub messaging pattern where publishers (senders) send messages without specifying direct receivers. **In this context, senders and receivers operate independently of each other, and subscribers' express interest in specific classes of messages**. In contrast in the present invention, the "system state manager" plays an active role in identifying a suitable "compute node" based on the attributes of the service request. In the present invention, communication is directed and purposeful, where the system state manager actively determines the appropriate compute node based on specific attributes of the service request. Further, D5 describe message filtering in pub/sub systems, where subscribers receive a subset of total published messages. **Filtering is based on topics or content**. The present invention, on the other hand, involves the system state manager **routing a specific service request to a compute node based on its ability and interest list configuration**.

As detailed in the Case Law of the Boards of Appeal at I.D.3.4.3:

> In T570/91 the board emphasized that although a person skilled in the art was completely free in choosing a starting point, he would of course be bound afterwards by that choice. If, for instance, the skilled person preferred and decided to start from a specific compressor piston, he could further develop that piston but at the end of that development the normal result would still be a compressor piston and not an internal combustion engine piston. In T 439/92 it was explained that a conscious choice of starting point, made in the knowledge of the respective benefits and drawbacks of the various types concerned, not only determined the subject-matter serving as a starting point but also defined the framework for further development, i.e. a further development within this particular type. A change of type during the further development of the consciously chosen type, to another type, which was previously known but had not been chosen, could then only be seen as the result of an ex-post-facto analysis (see a/so T 1040/93, T 35/95, T 739/95, T 255/03). It is unlikely, and normally not obvious, for the invention type originally chosen to be changed during

> development (T 817/94, recently cited in T 749/11 and T 535/10). A generically different document cannot normally be considered as a realistic starting point for the assessment of inventive step (T 870/96, T1105/92, T 464/98).

The propose of systems disclosed in D5 is to limit the flexibility in modifying both the publisher and the structure of the published data. In contrast, the present invention provides a system that dynamically considers interest configurations of nodes for efficient service request execution, which goes beyond the generic publish-subscribe messaging pattern discussed in D5.

Thus, D5 lacks any teaching of system state managers' dynamic evaluating attributes to select a compute node based on **ability and interest list configuration**. Therefore, senders as disclosed in D5 cannot be presumably equated to a "system state manager" and the "receivers" as disclosed in D5 to a "compute node" that has expressed a corresponding interest as claimed in the amended independent claim 1.

In addition, the information provided in D5, which discusses the publish-subscribe messaging pattern in software architecture, does not suggest or teach about distributed compute and storage as outlined in the present invention. D5 primarily focuses on the dynamics of message exchange between publishers and subscribers, the use of intermediaries like message brokers, and the concept of topics and content-based filtering. However, in contrast, the present invention involves the integration of distributed compute and storage, specifically mentioning a super state structure representing metadata associated with service requests or the output of execution. While D5 discusses the general principles of message-oriented middleware and the interaction between publishers and subscribers, it does not delve into the detailed architecture of distributed compute coupled with distributed storage, especially in the context of managing service requests and their associated metadata. Therefore, one cannot infer or derive the claimed features of the present invention from the information provided in D5.

Applicant respectfully submits that the focus of D5 on message patterns and brokers cannot be extended to the specific configuration of distributed compute and storage as outlined in the claimed invention. The present invention is directed towards the orchestration of computing resources and the storage of metadata, ensuring that the distributed compute is operatively and communicatively coupled with distributed storage, which is not taught or suggested in D5.

D5 at best describe facilitating a direct exchange of metadata between publishers and subscribers, utilizing IP multicast. This shared metadata forms the basis for decentralized message routing within the DDS middleware. However, nowhere does D5 teaches or suggests that senders (presumably equated to system state manager) is configured to store a super state structure (306) representing metadata associated with said service request or output of said execution of said service request by said identified compute node (404). In fact, D5 is completely silent on storing a super state structure representing metadata associated with said service request or output of said execution of said service request. Therefore, Applicant respectfully submits that D5 fails to teach or suggest the above- mentioned features as recited in amended independent claim 1.

When all the distinguishing features of disclosures of D5 are taken into account, and the objective technical problem formed according to the problem and solution approach, then it is very clear that a skilled person would not arrive at the present invention given the disclosures of D5.

Accordingly, D5 fails to disclose "*a system state manager (302) that, based on at least one attribute of each service request in the interaction pool, identifies a compute node (404) from one or more available compute nodes comprised in the distributed compute (304), that has an ability and an interest list configuration to execute said respective service request, and transmits said respective service request to said identified compute node (404) for execution, wherein said interest list configuration indicates an asset class of service requests that said identified compute node is configured to execute, and wherein said distributed compute (304) is operatively and communicatively coupled with a distributed storage (308), said distributed storage (308) being configured to store a super state structure (306) representing metadata associated with said service request or output of said execution of said service request by said identified compute node (404)", as recited in amended independent claim 1 (emphasis added).*

Further, D1-D4 have fails to cure the deficiencies of D5 to teach or suggest the features of amended independent claim 1.

Technical effect

One of the problems that the claimed systems and methods seek to solve is that "existing node-based computing architectures are **agnostic of the interest of the node** (N) i.e. **the service requests that the node wishes to execute/process**" as described in paragraph [0004] of the as-filed specification. In other words, the existing architectures, systems, and methods do not take into account any interest configuration (i.e., type of transaction or asset class of service requests) of the nodes.

The claimed system address this problem by "*a system state manager (302) that, based on at least one attribute of each service request in the interaction pool, identifies a compute node (404) from one or more available compute nodes comprised in the distributed compute (304), that has an ability and an interest list configuration to execute said respective service request, and transmits said respective service request to said identified compute node (404) for execution, wherein said interest list configuration indicates an asset class of service requests that said identified compute node is configured to execute, and wherein said distributed compute (304) is operatively and communicatively coupled with a distributed storage (308), said distributed storage (308) being configured to store a super state structure (306) representing metadata associated with said service request or output of said execution of said service request by said identified compute node (404)".*

One of the technical effects associated with the amended independent claim 1 is that the claimed system can orchestrate nodes in a dynamic manner and facilitate high frequency operations. Further the system as claimed facilitate selective hearing in order to form a state channel that can verify respective transactions belonging to a specific asset class relatively ahead of others as described in [00138] of the as-filed specification. In addition, the claimed system provides a feature wherein a node owner (of the compute node) may specify (in the interest list) any number of transaction types of assets classes that the node owner is interested in the compute node to process. This feature has been claimed in amended claims 8 and 12.

Thus, D1-D5 either alone or in combination do not disclose the solution as claimed in amended independent claim 1, thereby rendering claim 1 novel and inventive over D1, D2, D3, D4 and D5.

Amended independent claim 11 recites subject matter analogous to amended independent claim 1. Therefore, the remarks presented above for amended independent claim 1, also apply to

amended independent claim 11, which renders amended independent claim 11 novel and inventive.

Dependent claims

The dependent claims are also therefore novel and inventive at least by virtue of their dependency on amended independent claim 1, which should be allowable for the reason stated above.

SUMMARY

Applicant submits that the application now complies with the provisions of the act and requests that the Examiner passes the application through to grant.

Before any adverse decision is made concerning the patentability or procedural aspects of the above application, we request a personal or telephone interview with the Examiner, or, if the Examiner will not grant an interview, oral proceedings.

Yours faithfully,

Attachments:
Amendments to the claims – Marked up Copy
Amendments to the claims – Clean Copy