Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements

draft-ietf-cats-usecases-requirements-06

K. Yao, China Mobile
LM. Contreras, Telefonica
H.Shi,Huawei
S. Zhang, China Unicom
Qing An, Alibaba Group

Updates Since the Interim Meeting

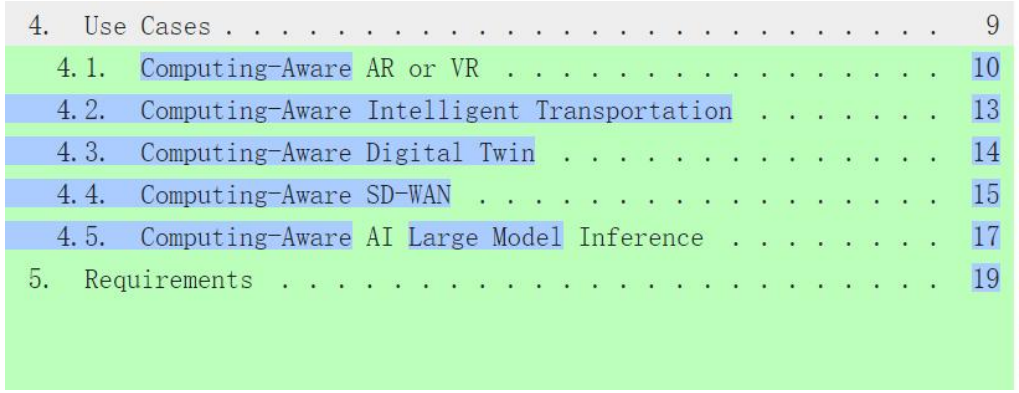
- **There are two major updates since CATS interim meeting on Jan 15th.**
- **From -04 to -05 :**
  - Addressed comments from Cheng Li https://mailarchive.ietf.org/arch/msg/cats/QRpZNZwZeHMePlhH21YCkR5u9iI/
  - Updated requirements (Address comments from Cheng Li on Sec 5.1 and Sec 5.4, contributions from Jordi on Sec 5.3)
  - Updated security considerations(contributions from Meiling)
- **From -05 to -06:**
  - Revised use cases (contributions from Dirk, Jordi, and Paul)
  - Added correlation between use cases and requirements in section 5.6 (Address comments from Joel, Julien, and Dirk)

Updates Since the Interim Meeting

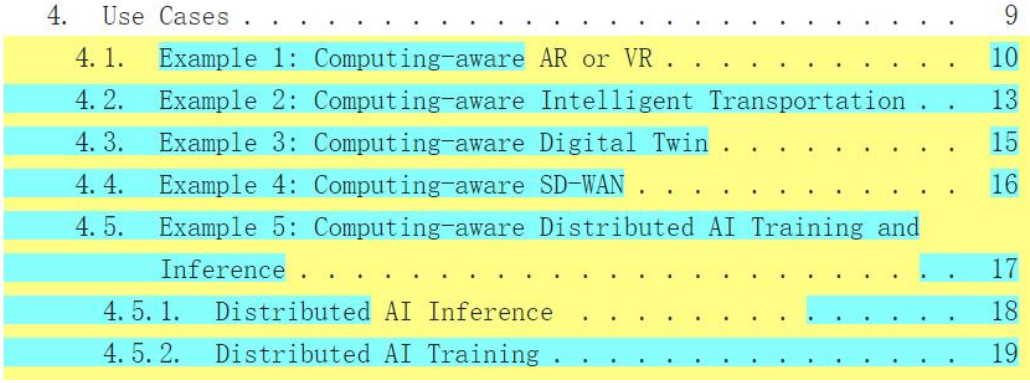
- **Progress of the four individual documents presented in the interim meeting:**
  - draft-jeong-cats-its-use-cases (**Part of which has been integrated into the WG document, in section 4.2**)
  - draft-ftzhs-cats-industrial-requirement (**The recent update in the document seems not raising new CATS requirements, will see new progress.**)
  - draft-lcmw-cats-midhaul (**Authors consider to go separately for architectural aspects**)
  - draft-wang-cats-green-challenges & draft-wang-cats-usecase-green (**Part of the considerations on energy and cost metrics has been integrated into the WG document, in section 4.4 & section 4.5**)

Updates on Use Cases

- There are five use cases in section 4



draft-ietf-cats-usecases-requirements-04



draft-ietf-cats-usecases-requirements-06

- Deleted some redundant description in section 4.1
- Merged revision on section 4.2 from Paul and his draft **draft-jeong-cats-its-use-cases-04**
- Added some considerations on cost and energy metrics to sec 4.4 from Dirk.
- Splitted section 4.5 into **distributed AI inference** and **distributed AI training**.
  - **4.5.1 Distributed AI inference**, Modify figure 5(delete the central cloud part). Put more emphasis on the delay, compute resource and cost metrics at edge site.
  - **4.5.2 Distributed AI training,** a newly added part(contribution from Dirk and Jordi) where storage and cost metrics are important for CATS.

Updates on Requirements

- **Requirements on Instance Selection(Sec 5.1)**
  - R2 (in -04): MUST provide an mapping methods for further quickly selecting the service instance. **(DELETED)**
  - R3 (in -04): SHOULD provide a timeout limitation for selecting the service instance. **(DELETED)**
  - R5 (in -04): MUST provide a mechanism for solving the service contention problem when multiple service instances with the same service identifier are all available to provide computing services. **(DELETED)**

Updates on Requirements -- Cont'd

- **Requirements on Metric Definition and Representation (Sec 5.2)**
  - Merged sec 5.2 (Support Agreement on Metric Representation) and part of requirements of sec 5.4 (Support Alternative Definition and Use of Metrics) in version -04
  - R7 (in -04): MUST include network metrics. **(DELETED)**
  - R4 (in -06): A model of the compute and network resources MUST be defined. Such a model MUST characterize how metrics are abstracted out from the compute and network resources. We refer to this model as the Resource Model. **(REPLACED R12 in -04)**
  - R5 (in -06): The Resource Model MUST be implementable in an interoperable manner. That is, independent implementations of the Resource Model must be interoperable. **(NEW)**

Updates on Requirements -- Cont'd

- **Requirements on Metric Definition and Representation (Sec 5.2)**
  - R6 (in -06): The Resource Model MUST be executable in a scalable manner. That is, an agent implementing the Resource Model MUST be able to execute it at the required time scale and at an affordable cost (e.g., memory footprint, energy, etc.). **(NEW)**
  - R7 (in -06): The Resource Model MUST be useful. That is, the metrics that an agent can obtain by executing the Resource Model must be useful to make node and path selection decisions. **(NEW)**
  - R8 (in -06): There MUST set up metric information that can be understood by CATS components. For metrics that CATS components do not understand or support, CATS components will ignore them. **(REPLACED R14 in -04)**
  - R9 (in -06): The computing metrics in CATS MUST be simple, that is distributing metrics and selecting path based on these metrics will not cause routing loops and route oscillation. **(NEW)**

Updates on Requirements -- Cont'd

- **Requirements on Use of CATS Metrics (Sec 5.3)**
  - Merged sec 5.3 (Support Moderate Metric Distributing) and part of requirements of sec 5.4 (Support Alternative Definition and Use of Metrics) in version -04
  - R11 (in -06): MUST declare the entity that collect metrics. **(NEW)**
  - R11 (in -04): MUST realize means for rate control for distributing of metrics. **(DELETED)**
  - R13 (in -04): In addition to common metrics that are agreed by all CATS components like processing delay, there SHOULD be some other ways for metrics definition, which is used for the selection of specific service instance. **(DELETED)**
  - R14 (in -06): MUST be clear of the update frequency of CATS metrics and its corresponding distribution method. **(NEW)**
  - R15 (IN -06): SHOULD provide mechanisms to assess selection accuracy and re-select metrics if the selection result is not accurate. **(NEW)**

Updates on Requirements -- Cont'd

- **Requirements on Instance Affinity (Sec 5.4)**
  - R16 (in -06): Instance affinity MUST be maintained when the transaction is stateful. **(REPLACED R15 in -04, previous:** Instance affinity MUST be maintained when state information is needed. **)**
  - R17 (in -06): Instance affinity MUST be maintained for service requests or transactions that belong to the same flow. **(REPLACED R16 in -04, previous:** MUST maintain instance affinity which MAY span one or more service requests, i.e., all the packets from the same application-level flow MUST go to the same service instance unless the original service instance is unreachable **)**
  - R18: MUST avoid keeping fine runtime-state granularity in network nodes for providing instance affinity. For example, as mentioned above, maintaining per-flow states for a specific APP. **(REPLACED R17 in -04, previous:** MUST avoid keeping fine runtime-state granularity in network nodes for providing instance affinity. **)**

Updates on Requirements -- Cont'd

- **Sec 5.6 Correlation between Use Cases and Requirements (NEW)**
  - Please Check figure 6 in version -06 (too long to copy here)
  - All five use cases in section 4 need requirements from R1 to R18, and R21
  - R19: MUST provide mechanisms to minimize client side states in order to achieve the instance affinity. **(Only for sec 4.1 & 4.2)**
  - R20: SHOULD support the UE and service instance mobility. **(Only for sec 4.2)**

Use Cases and Requirements from Other I-Ds

Integrated Sensing and Communications (ISAC) Use Case for CATS

draft-bernardos-cats-isac-uc, Carlos

CATS Requirements Considerations for Service Segmentation-related Use Cases

draft-dcn-cats-req-service-segmentation, Minh-Ngoc

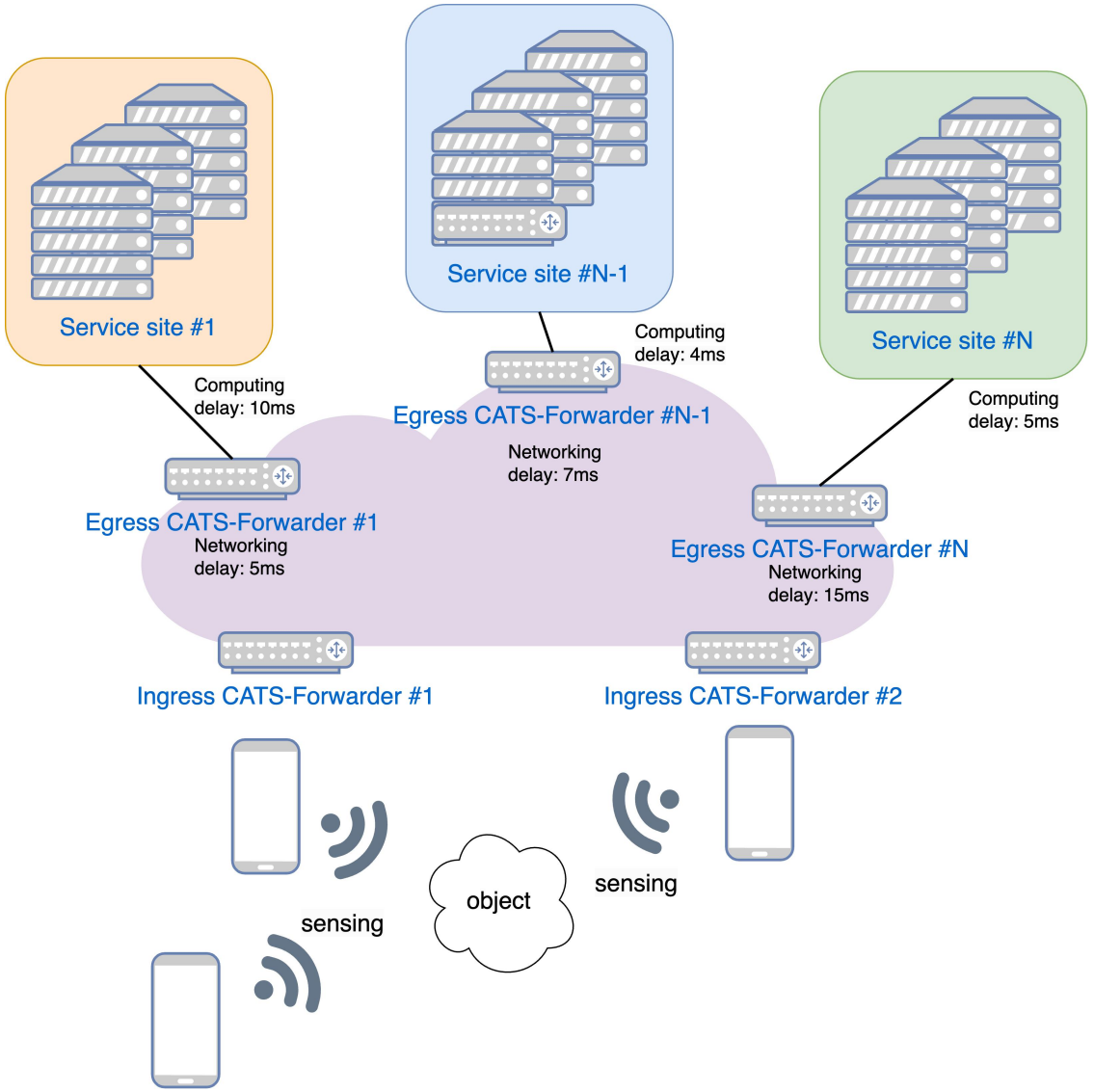
Integrated Sensing and Communications (ISAC) use case for CATS

draft-bernardos-cats-isac-uc-00

Carlos J. Bernardos (UC3M)
Alain Mourad (InterDigital)

March 2025

ISAC CATS UC: use case description



- Integrated Sensing and Communications (ISAC) enables wireless networks to perform simultaneous data transmission and environmental sensing
- Multiple network nodes collect raw sensing data
- Collected raw data must be aggregated and processed by a designated computing node with sufficient computational resources
- UC aligned with ETSI ISAC ISG WI#5 ('Integration of Computing with ISAC’)

ISAC CATS UC: relation to CATS

- In the distributed sensing use case, the sensed data collected by multiple nodes must be efficiently routed to a computing node capable of processing it
  - The choice of the computing node depends on several factors, including computational load, network congestion, and latency constraints
  - CATS mechanisms can optimize the selection of the processing node by dynamically steering the traffic based on computing resource availability and network conditions.
- As sensing data is often time-sensitive, CATS can ensure low-latency paths while balancing computational demands across different processing entities
  - This capability is essential for real-time applications such as cooperative perception for autonomous systems, industrial monitoring, and smart city infrastructure

ISAC CATS UC: requirements

- **Traffic Steering and Resource Allocation:** Ensuring that sensing data is directed to the most suitable computing node while considering both network conditions and processing availability.
- **Latency Sensitivity:** Many ISAC applications require near-real-time processing, necessitating low-latency and high-reliability data forwarding strategies
- **Data Synchronization:** Sensing nodes may have different perspectives on the environment, requiring synchronization and fusion of data streams before processing
- **Scalability:** As the number of participating sensing nodes increases, mechanisms must efficiently distribute and balance the computational workload
- **Security and Privacy:** Sensed data may contain sensitive information, requiring mechanisms for secure transmission and processing

ISAC CATS UC: discussion

- ISAC UC (and associated requirements) not covered in draft-ietf-cats-usecases-requirements-06

- Proposal is to include (some) text from draft-bernardos-cats-isac-uc-00 into draft-ietf-cats-usecases-requirements-06

- New WI#5 of ETSI ISAC ISG is expected to be very much aligned with IETF CATS discussion (maybe some kind of liaison is desirable at ETSI ISAC ISG – IETF CATS WG level)

Additional CATS requirements Consideration for Service Segmentation-related use cases

draft-dcn-cats-req-service-segmentation-00

Minh-Ngoc Tran (Soongsil University), Younghan Kim (Soongsil University)

This Document's Target

- **Service Segmentation[1]** is an service deployment option
  [1]draft-li-cats-task-segmentation-framework
  - A Service **is split into multiple subtasks** which can run in different computing sites.
  - A Service request is served by the **combination of these subtasks.**
  - These subtasks can run **in parallel or in sequence/chain.**

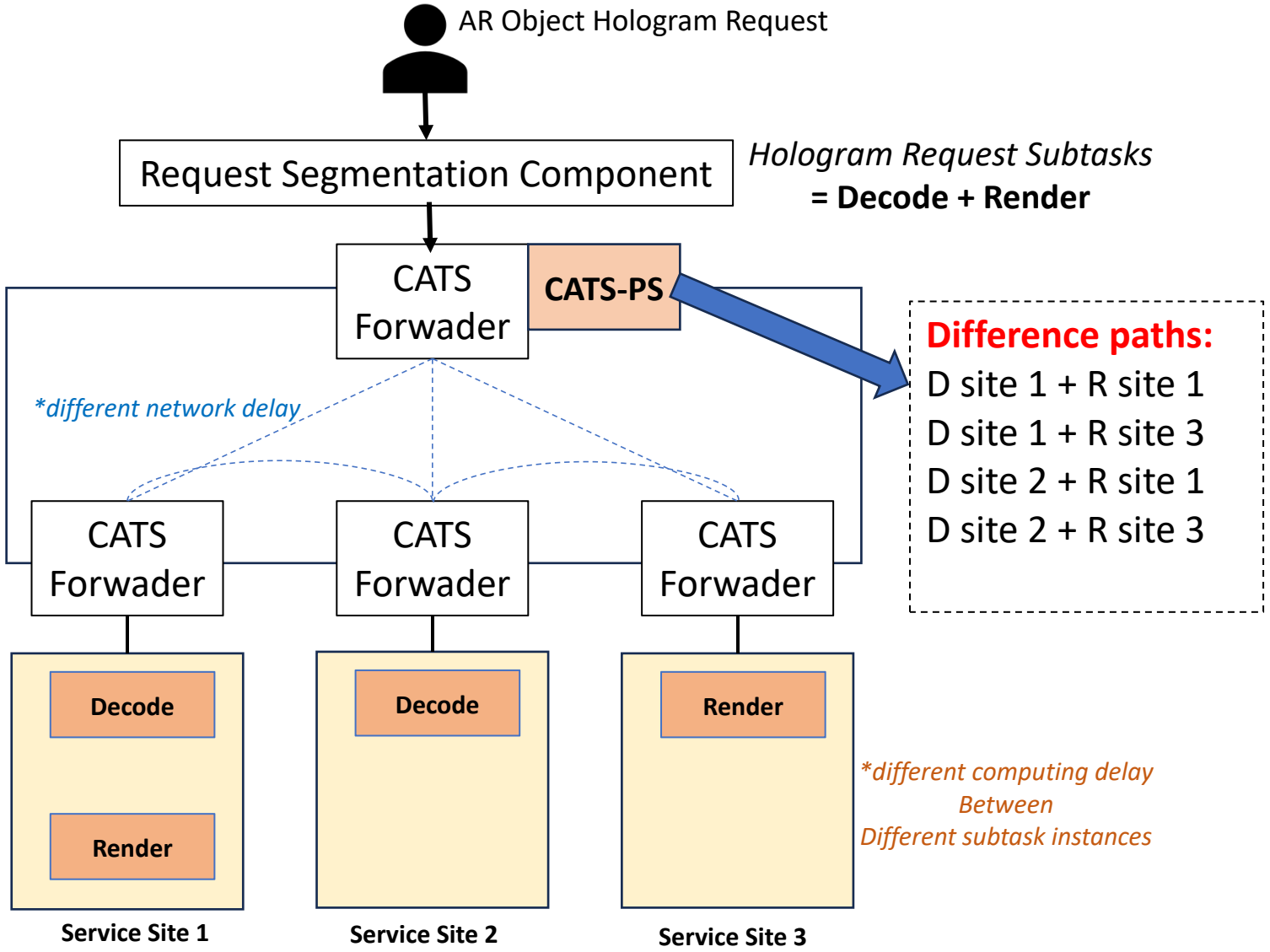
- **2 Use Cases** in CATS Use Cases I-D (draft-ietf-cats-usecases-requirements) **have existing examples where service segmentation is applied:**
  (current UC I-D has not covered Service Segmentation deployment option in these 2 use cases)
  - AR-VR
    - Example: AR Hologram Subtask Sequence
  - Distributed AI Training
    - Example: Hierarchial Federated Learning

- **This document discusses:**
  - Differences that Service Segmentation causes to CATS system design in these CATS Use Cases
  - Based on that, propose additional CATS requirements

  Not a separated Use Case, but new requirement if Existing Use Cases apply Service Segmentation

Differences with normal CATS

Example 1: AR Hologram Subtask Sequence

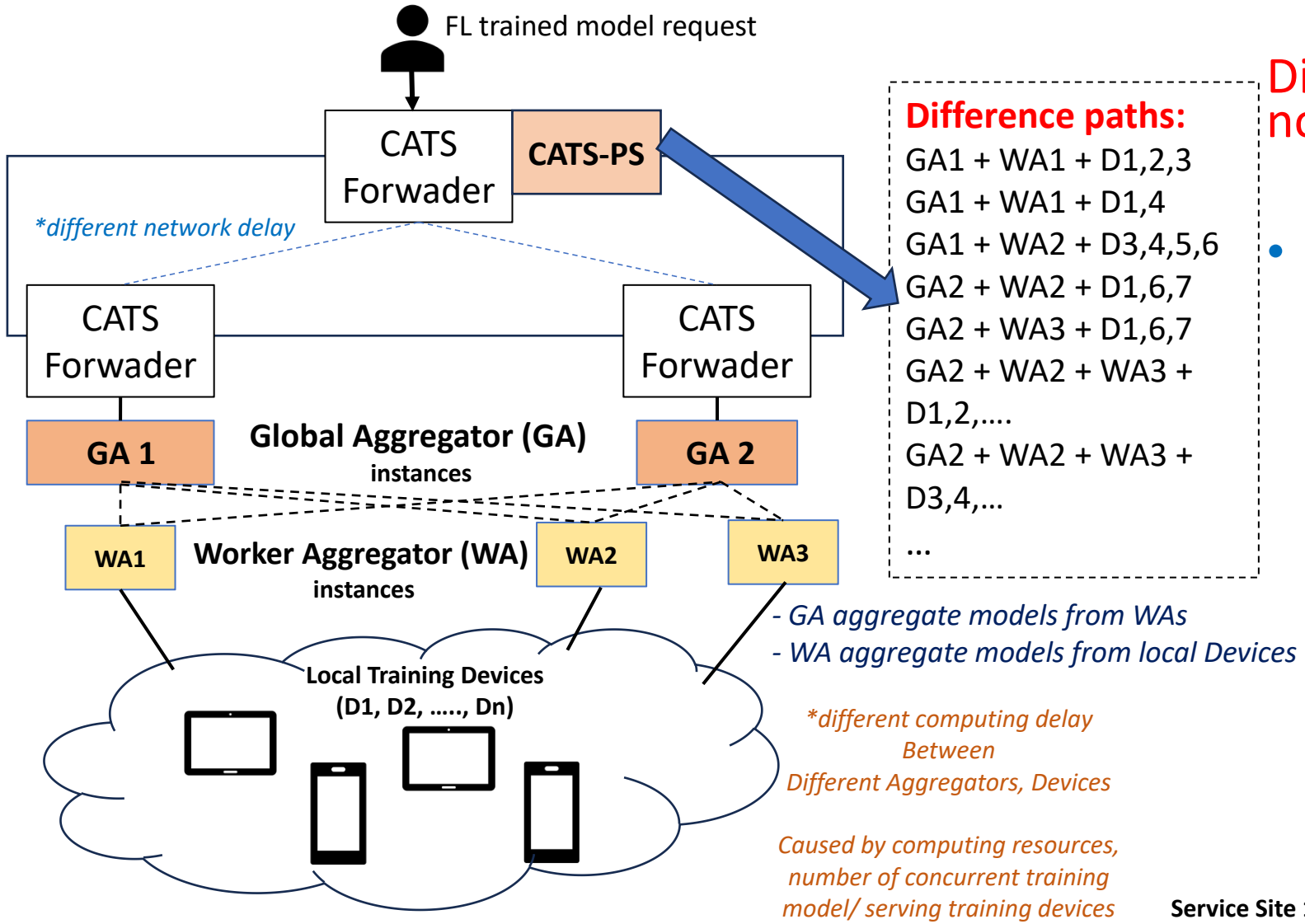


Differences with normal CATS case – no service segmentation

- Select an optimal subtask combination (subtask sequence/chain) instead of select an optimal instance
  - Each subtask can have multiple instances
  - Each subtask instance have different performance because of its associated computing/network resource/request serving status
  - Different subtask sequences/chains have different performance/CATS cost

Differences with normal CATS

Example 2: Hierarchial Federated Learning



Difference paths:

GA1 + WA1 + D1,2,3
GA1 + WA1 + D1,4
GA1 + WA2 + D3,4,5,6
GA2 + WA2 + D1,6,7
GA2 + WA3 + D1,6,7
GA2 + WA2 + WA3 + D1,2,….
GA2 + WA2 + WA3 + D3,4,…
…

- GA aggregate models from WAs
- WA aggregate models from local Devices

**different computing delay Between Different Aggregators, Devices*

Caused by computing resources, number of concurrent training model/ serving training devices

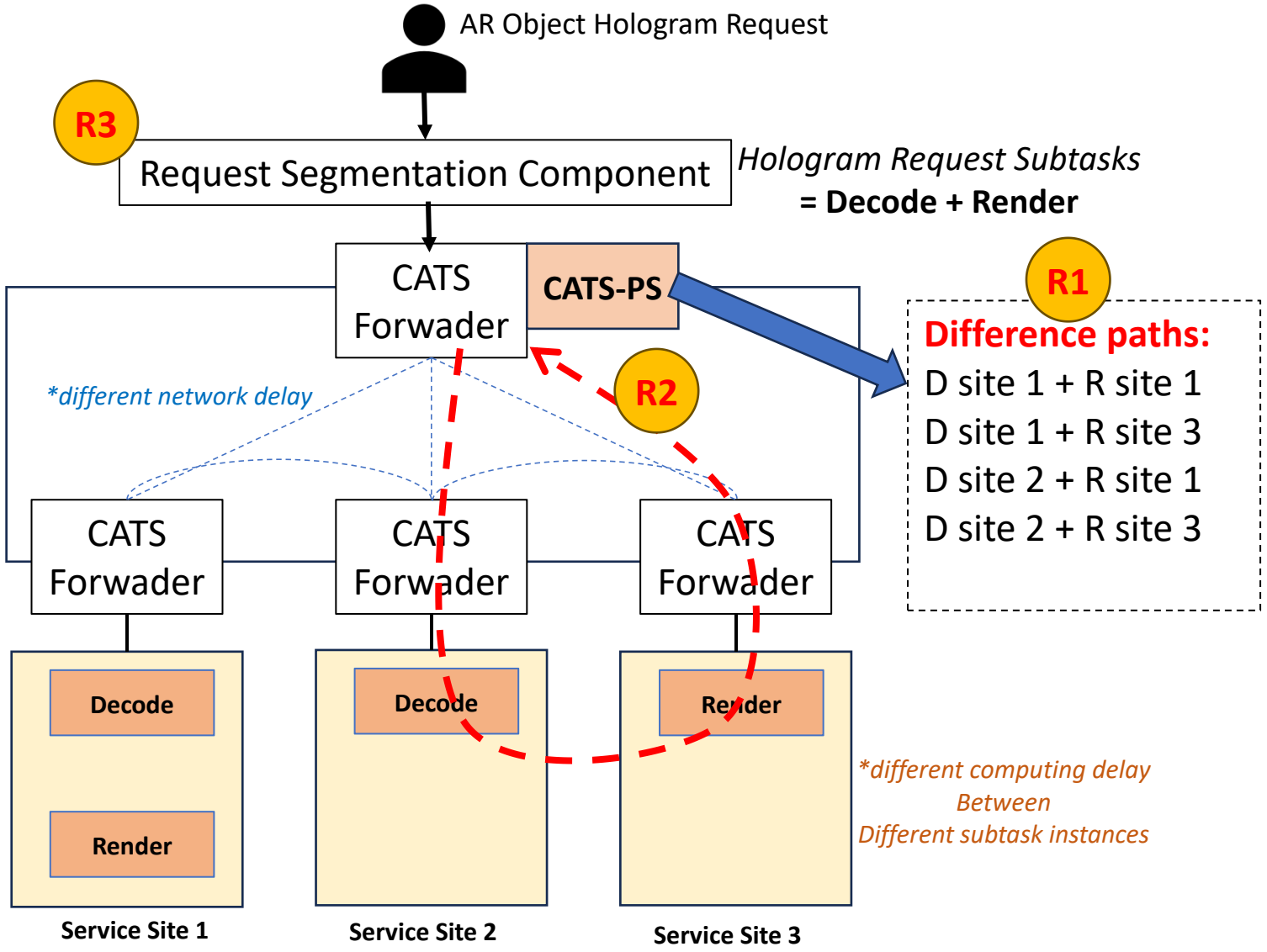
Service Site 1

Differences with normal CATS case – no service segmentation

- Select an optimal subtask combination (aggregators + devices) instead of select an optimal instance
  - Each subtask can have multiple instances
  - Each subtask instance have different performance because of its associated computing/network resource/request serving status
  - Different subtask combination have different performance/CATS cost

Requirements

Example 1: AR Hologram Subtask Sequence

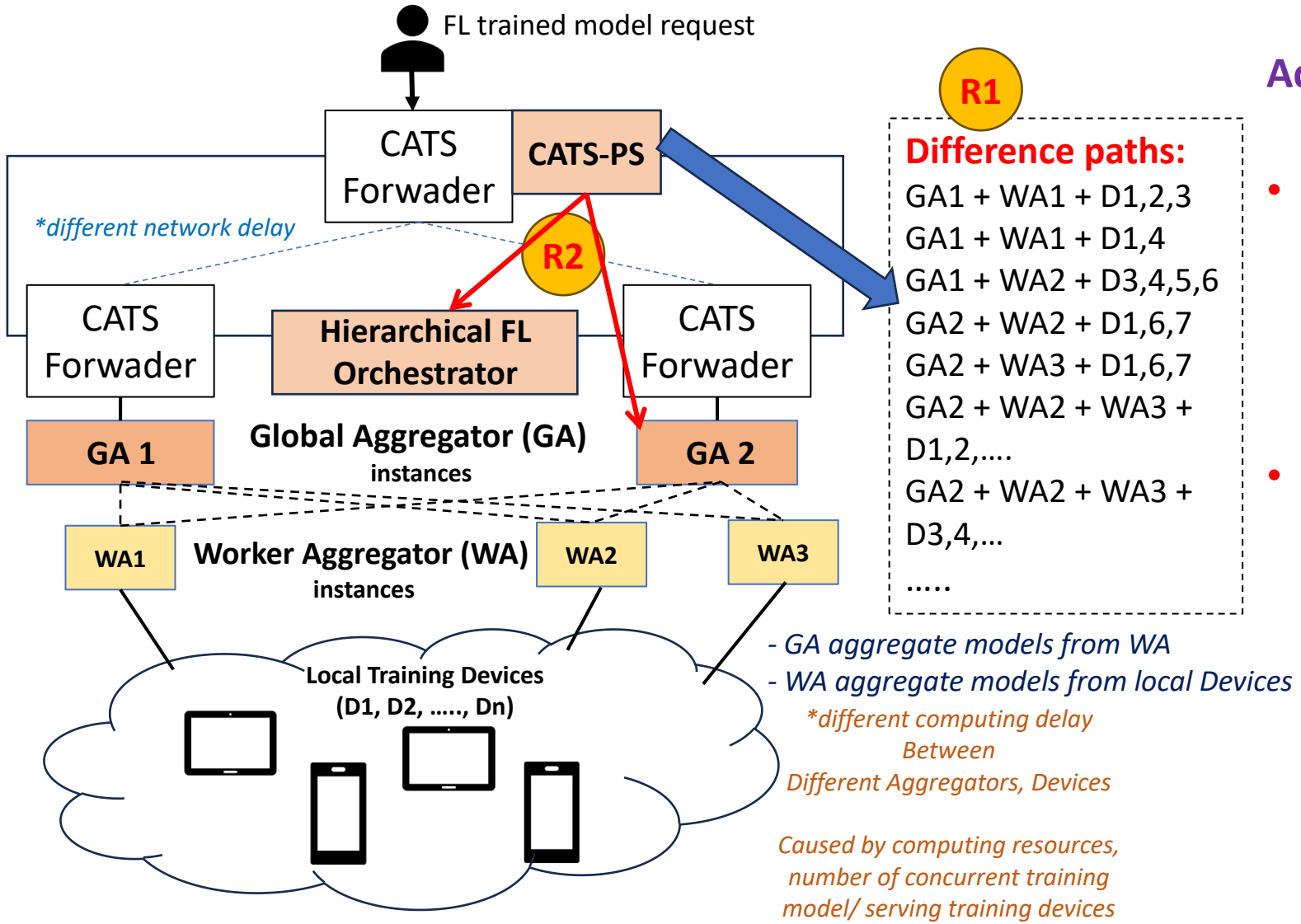


Additional CATS requirements

- **R1:** CATS metrics and metric aggregation should
  - Consider each subtask computing/network capbalities
  - Distinguish different subtask combination (sequence/chain)
- **R2:** A mechanism that routes request to the determined optimal subtask sequence/chain
  - e.g Segment routing
- **R3:** When a full service instance is not existed, A mechanism that map service request to segmented subtask
  - E.g Task Segmentation Module in draft-li-cats-task-segmentation-framework

Requirements

Example 2: Hierarchial Federated Learning



R1

Difference paths:

GA1 + WA1 + D1,2,3
GA1 + WA1 + D1,4
GA1 + WA2 + D3,4,5,6
GA2 + WA2 + D1,6,7
GA2 + WA3 + D1,6,7
GA2 + WA2 + WA3 + D1,2,….
GA2 + WA2 + WA3 + D3,4,…
…..

- GA aggregate models from WA
- WA aggregate models from local Devices

**different computing delay Between Different Aggregators, Devices*

Caused by computing resources, number of concurrent training model/ serving training devices

Additional CATS requirements

- **R1:** CATS metrics and metric aggregation should
  - Consider each subtask computing/network capbalities
  - Distinguish different subtask combination (sequence/chain)
- **R2:** A mechanism that guides the FL system to use the determined optimal subtask combination
  - e.g C-PS request Hierarchical FL Orchestrator/ Global Aggregator

Additional CATS requirements

- Possibly add these Requirements to ***Section 5.1 Support dynamic and effective selection among multiple service instances***
- ***After R2: "MUST provide a method to determine the availability of a service instance."***
- Mentions these optional requirements below when Service Segmentation is used

- R2-1: A CATS system should consider subtask instance's computing and network resource condition and distinguish capabilities of different candidate combination of subtasks to serve a CATS service request.

- R2-2: A CATS system should provide mechanism to notice/guide/request the host computing entities of the services and service subtasks to implement the determined optimal sub-tasks combination.

- R2-3: A CATS system should provide mechanism to map the service request to corresponding segmented subtasks if the original service is not existed, only subtask instance endpoints are available.

Open questions

- **Should we prioritize CATS use cases?(Question from Julien)**
  - 1. AR/VR; 2. ITS; 3. Digital twin; 4. SD-WAN; 5. Distributed AI;
  - 6. CDN (**in the appendix, will be deleted in the final version**); 7. Cloud rendering (**in the appendix**);
  - 8. O-RAN case, draft-lcmw-cats-midhaul-02, (**Authors consider to go separately for architectural aspects**);
  - 9. 3GPP-edge, draft-jiang-cats-usecase-5gedge-00, (**Same as #8**);
  - 10. Energy, draft-wang-cats-usecase-green (**Part of the considerations been merged in sec 4.4 and 4.5**);
  - 11. Industrial internet, draft-ftzhs-cats-industrial-requirement-03 (**Decision deadline before IETF123**)
  - 12. Integrated Sensing and Communication(ISAC) (**Decision deadline before IETF123**);
  - 13. Service Segmentation-related use cases (**Decision deadline before IETF123**)
  - Others

Principles from Chairs

- How significantly different is your use case from those already described in the WG draft? Would it be possible to include your use case as an example of one of the other major use cases?
- Look at the requirements in the WG document. Does your use case open up any additional requirements? If so, let's know about them and discuss how to make sure they are described.
- Look at the framework document. Can you see how your use case would be addressed with the framework? If not, are there changes you can propose?

Next Steps

- Polish existing document and check if any new use cases or requirements can be merged
- Address comments from Paul
  - https://mailarchive.ietf.org/arch/msg/cats/6KjRWVcUcTIRNZ_JI0op0QEeKjc/
- Address comments from Julien over instance affinity related requirements
- **When ready, authors request for publication of this document as an informational RFC (currently tagged as information-only)**
  - **Reason 1: Document maturity.** This document include some reasonable use cases, and the requirements are implementatble. Authors also have provided some PoC results in the appendix.
  - **Reason 2: Topic popularity.** There are some ongoing discussions over computing issues inside and outside of routing area in IETF. This document can be helpful for understanding the scope and problems.
  - **Reason 3: Value of the solution.** Other SDOs like 3GPP, ETSI, ORAN, and ITU-T all have discussions and ongoing work over CATS related topics. There maybe applicability of CATS inside these SDOs. This document can serve as a good reference for these SDOs.

- **If it's agreed, authors plan to target the WGLC date at July 2025.**