CATS Problem Statement, Use cases and Requirements

draft-ietf-cats-usecases-requirements-01

K. Yao, China Mobile
D. Trossen，Huawei
M. Boucadair，Orange
LM. Contreras，Telefonica
H.Shi,Y.Li,Huawei
S. Zhang, China Unicom

Draft updates

Table of Contents

- **Introduction**
  - Rewrite part of the Introduction to avoid some verbose description.
- **Terminology**
  - Echo terminologies updates from the architecture draft: *draft-ldbc-cats-framework-03*
- **Use cases**
  - Will merge the Computing-Aware AI large model use case after this meeting.
- **Requirements**
  - Updates on requirements of service and instance selection.
  - Updates on metrics definition.
  - Others.

Terminologies:

- **Service:** An offering that is made available by a provider by orchestrating a set of resources (networking, compute, storage, etc.). Which and how these resources are solicited is part of the service logic which is internal to the provider. For example, these resources may be:
  * Exposed by one or multiple processes (a.k.a. Service Functions (SFs)). [RFC7665]
  * Provided by virtual instances, physical, or a combination thereof.
  * Hosted within the same or distinct nodes.
  * Hosted within the same or multiple service sites.
  * Chained to provide a service using a variety of means.

  How a service is structured is out of the scope of CATS. The same service can be provided in many locations; each of them constitutes a service instance.
- **Service identifier**: An identifier representing a service, which the clients use to access it.
- **Computing service:** An offering that is made available by a provider by orchestrating a set of computing resources (without networking resources).
- **Service instance:** An instance of running resources according to a given service logic. Many such instances can be enabled by a provider. Instances that adhere to the same service logic provide the same service. An instance is typically running in a service site. Clients' requests are serviced by one of these instances.
- **Service site:** A location that hosts the resources that are required to offer a service. A service site may be a node or a set of nodes. A CATS-serviced site is a service site that is connected to a CATS-Forwarder.

Terminologies:

- **Network Edge:** The network edge is an architectural demarcation point used to identify physical locations where the corporate network connects to third-party networks.

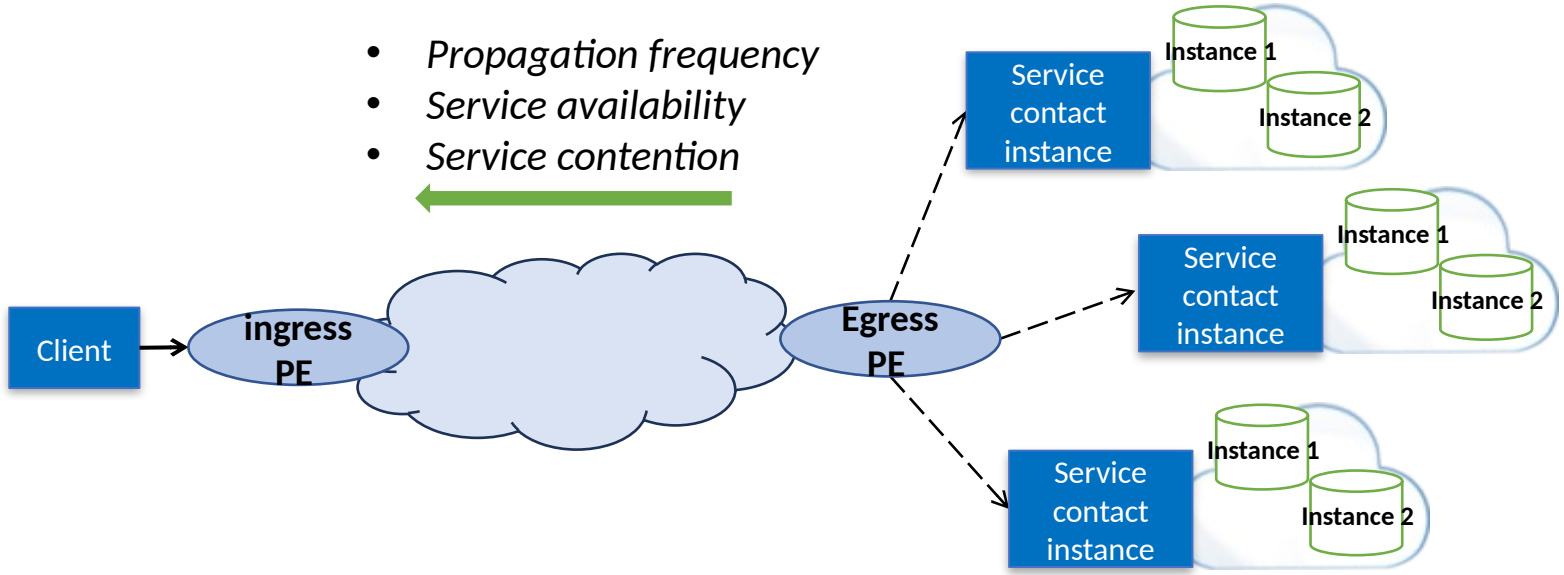
- **Edge Computing:** Edge computing is a computing pattern that moves computing infrastructures, i.e, servers, away from centralized data centers and instead places it close to the end users for low latency communication.

Relations with network edge: edge computing infrastructures connect to corporate network through a network edge entry/exit point.

Requirements:

➢ Support dynamic and effective selection among multiple serivce instances:

- R1: **MUST** provide a discovery and resolving methodology for the mapping of a service identifier to a specific address.
- R2: **MUST** provide an mapping methods for further quickly selecting the service instance.
- (Added)R3: **SHOULD** provide a timeout limitation for selecting the service instance.
- (Added)R4: **MUST** provide a method to determine the availability of a service instance.
- (Added)R5: **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.



Requirements:

➢ Support Moderate Metric Distributing:

- (Added)R8: **MUST** provide mechanisms for metric collection.
- R9: **MUST** provide mechanisms to distribute the metrics.
- R10: **MUST** realize means for rate control for distributing of metrics.

Requirements:

➢ Support Alternative Definition and Use of Metrics:

- Previous sub-title is “Support Flexible Use of Metrics” , avoid vague expression “Flexible”.

- (Previous)R8: there **MUST** exist flexibility in term of metrics definition and utilization for the selection of service instance.

A temporary version, different voices on metric definition in the list

- (Changed to)R12: 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.

- (Previous) R10: **MUST** use network and computing metrics in a flexible way that includes a default action for the interoperation of network nodes which may or may not support the specific metrics.

- (Changed to) R14: **MUST** include a default action for the interoperation of network nodes which may or may not support the specific metrics.

Requirements:

➢ Support Instance Affinity:

- Previous sub-title is “Support Session and Service Continuity”

- (Previous)R11: session as well as service continuity **MUST** be maintained.
- (Changed to)R15: Instance affinity **MUST** be maintained when state information is needed.

- (Previous)R13: **MUST** avoid keeping fine runtime-state granularity in network nodes for providing session and service continuity.
- (Changed to)R17: **MUST** avoid keeping fine runtime-state granularity in network nodes for providing instance affinity.

“Session Continuity”, “Service Continuity”, “Session and Service Continuity” and “Session Persistence” ***are unified to*** *“Instance Affinity”*

Use cases:

Use Case of Computing-Aware AI large model (*To Be Merged*)

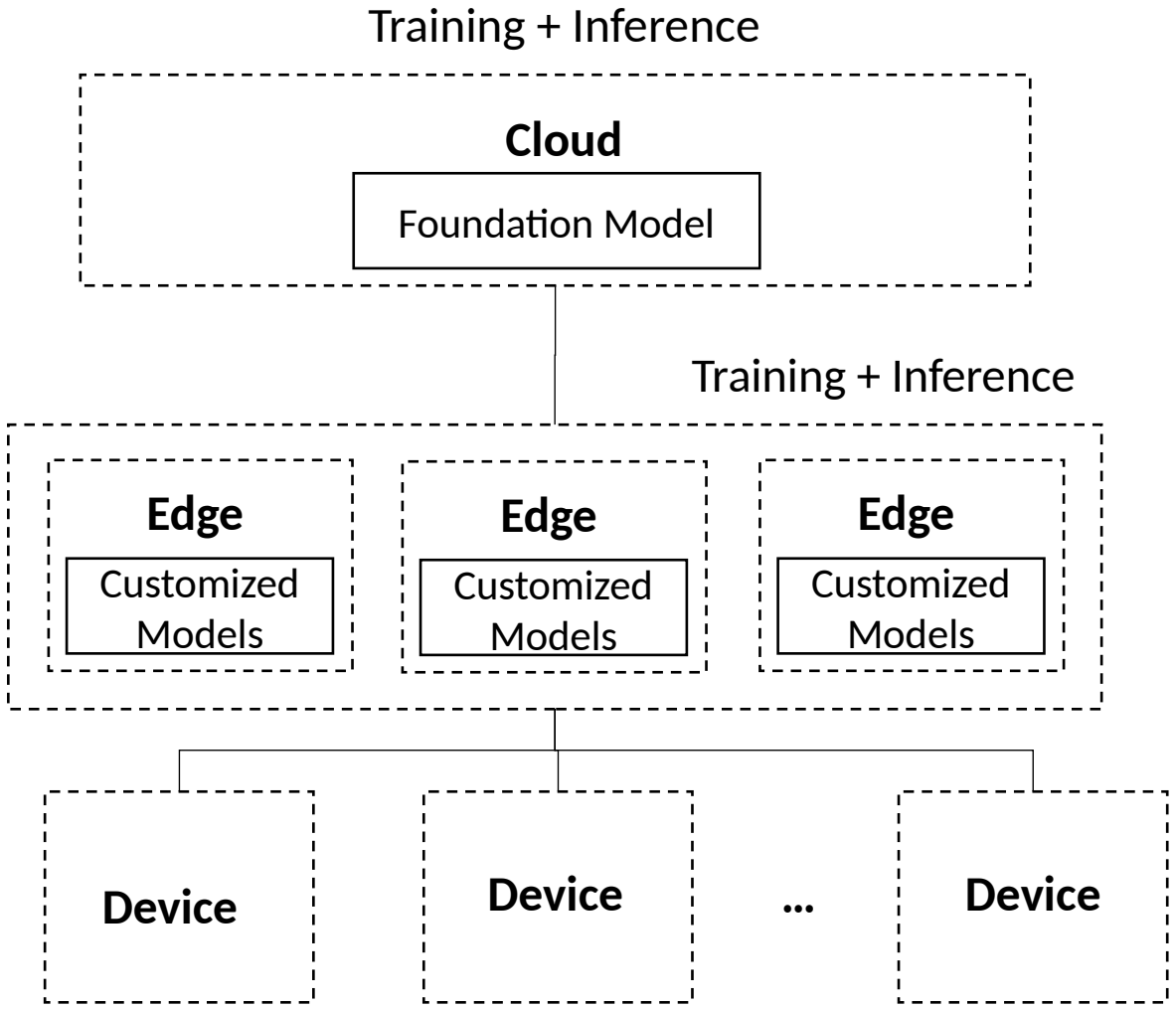
Contributions from ***Qing An(Alibaba Group)*** *anqing.aq@alibaba-inc.com*

- Draft: *https://datatracker.ietf.org/doc/draft-an-cats-usecase-ai/*
- Change Log after IETF 117: *https://github.com/cats-wg/draft-ietf-cats-usecases-requirements/pull/1*
- Focus more on CATS-specific matters
- Include three scenarios suitable for the CATS:
    1. Cloud-edge co-inference
    2. Cloud-edge-device co-inference
    3. Edge-alone inference (newly added)

Use cases – Computing-Aware AI large model:

- Cloud-edge co-inference

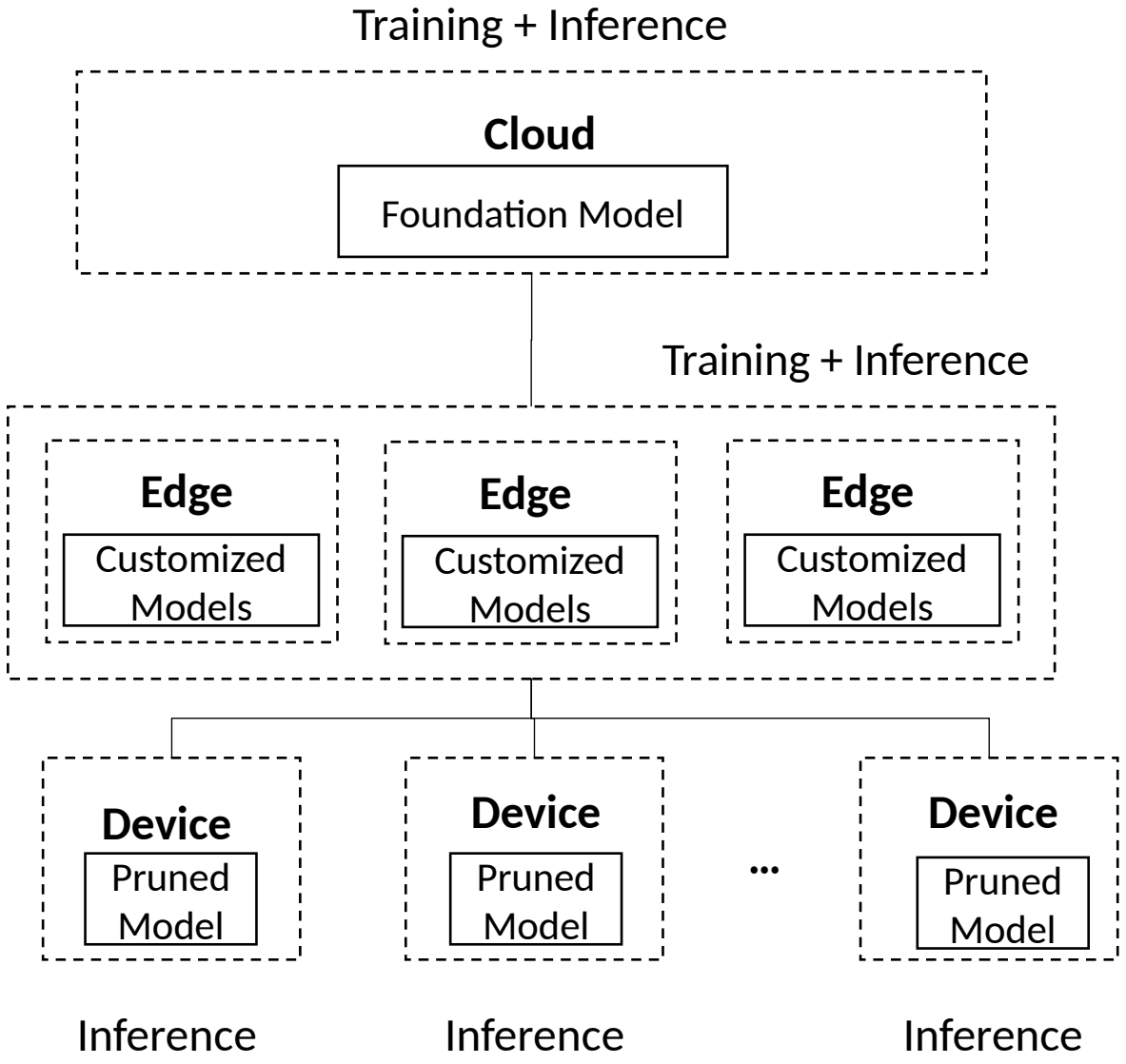
- Low latency: deploy inference near to device
- Low demand on device resources
- But when handling AI inference tasks, if traffic load between device and edge is high or edge computing resource is overloaded, traffic steering is needed to ensure the QoS



Use cases – Computing-Aware AI large model:

➢ Cloud-edge-device co-inference

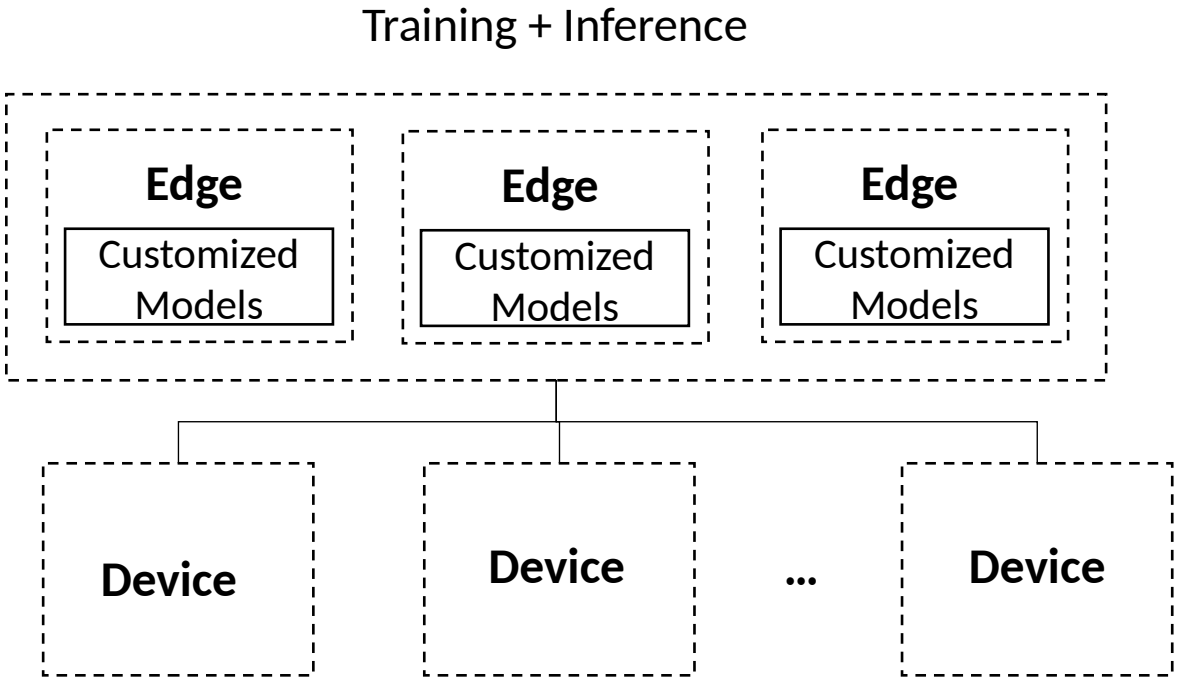
- More flexible deployment (also more complex): deploy inference locally or near to device
- Device can work when edge isn't available
- Careful consideration to ensure that edge will only be used when the trade-offs are right
- Similar to last scenario, traffic steering is needed



Use cases – Computing-Aware AI large model:

➢ Edge inference

- Edge can infer by its own without the cloud
- Low latency: deploy inference near to device
- Low demand on device resources
- Similar to last scenario, traffic steering is needed



Recap of Outstanding Comments Received in IETF 117

- some requirements are prescriptive, for example:
  - ruling out out-of-band methods with a “low latency” justification which does not clearly state why out-of-band is incompatible with low latency. (Our understanding is that the “low-latency” characteristic of in-band solution is intuitive, and it doesn’t mean that out-of-band method doesn’t work well.)
  - the “mapping of a service identifier to a specific address”. (We think this is required for service instance selection, since the same service ID may be bound to different addresses,but it also depends on how the CATS architecture is designed.)

- There are some vague expressions, for example:
  - “quickly” selecting an instance is not quantifiable so it is difficult to satisfy.(Add some requirements on quantification and other relative requirements.)
  - metrics flexibility is risky and could open up significant interoperability issues. (The wording has been modified but there is still on-going discussion on the scope of metric definition.)

- Metric collection is also an important aspect that should be considered. (The requirement is added.)

- Discussions on how session and service continuity are applied to different cases should be presented. Likely, service affinity is not required in all use cases. (Terms are unified to “Instance Affinity”.)

- Use cases are too high level to derive specific requirements.(Will merge the AI large model use case, but the correspondence between use cases and requirements need further updates.)

- Security considerations are too high level, and security requirements should be moved to section5, security considerations should talk about the security issues brought in by the mechanism.(Need further updates)

- Format issues.(Solved)

Next Steps:

- Update the correspondence between Use cases and the requirements.
- Keep the discussion on the requirement of metrics definition.
- Update security requirements and considerations.

Welcome for more comments and discussion.

https://github.com/cats-wg/draft-ietf-cats-usecases-requirements

Thanks!