Computing-Aware Traffic Steering
Internet-Draft
Intended status: Informational
Expires: 30 November 2026

B. Zhang, Ed.
Pengcheng Laboratory
Y. Dai, Ed.
Sun Yat-sen University
B. Shen, Ed.
W. Zhang, Ed.
Harbin Institute of Technology
Y. Qiao, Ed.
Pengcheng Laboratory

29 May 2026

Public Service Platform for Computing-Aware Traffic Steering (CATS)
draft-zhangb-cats-cmas-04

Abstract

CATS applications require service discovery and traffic steering across heterogeneous computing resources. Directly exposing raw computing metrics from different hardware platforms can be difficult for clients, service sites, and CATS control-plane components to interpret consistently. This Informational document describes a public service platform for CATS. The platform maintains a common service catalogue, associates public service identifiers with service descriptions and deployment requirements, and provides the service context used by service-oriented metric mechanisms. Service-oriented metric definitions and operational procedures are specified in [I-D.zhangb-cats-service-metrics-op-01].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 November 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

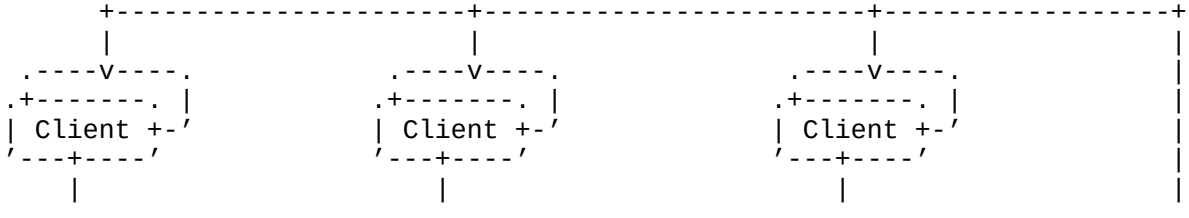
Computing-aware traffic steering (CATS) is a traffic engineering approach that takes into account the dynamic nature of computing resources and network state to optimize service-specific traffic forwarding towards a given service instance. As described in [I-D.ietf-cats-framework-24], the Computing-Aware Traffic Steering (CATS) framework assumes that there might be multiple service instances that are providing one given service, which are running in one or more service sites. Each of these service instances can be accessed via a service contact instance, which is a client-facing service function instance. A single service site may host one or multiple service contact instances. A single service site may have limited computing resources available at a given time, whereas the various service sites may experience different resource availability issues over time. Therefore, steering traffic among different service sites can address the issues of lacking resources in a specific service site. Based on this, [I-D.ietf-cats-framework-24] provides an architectural framework that aims at facilitating the making of compute- and network-aware traffic steering decisions in networking environments where computing service resources are deployed.

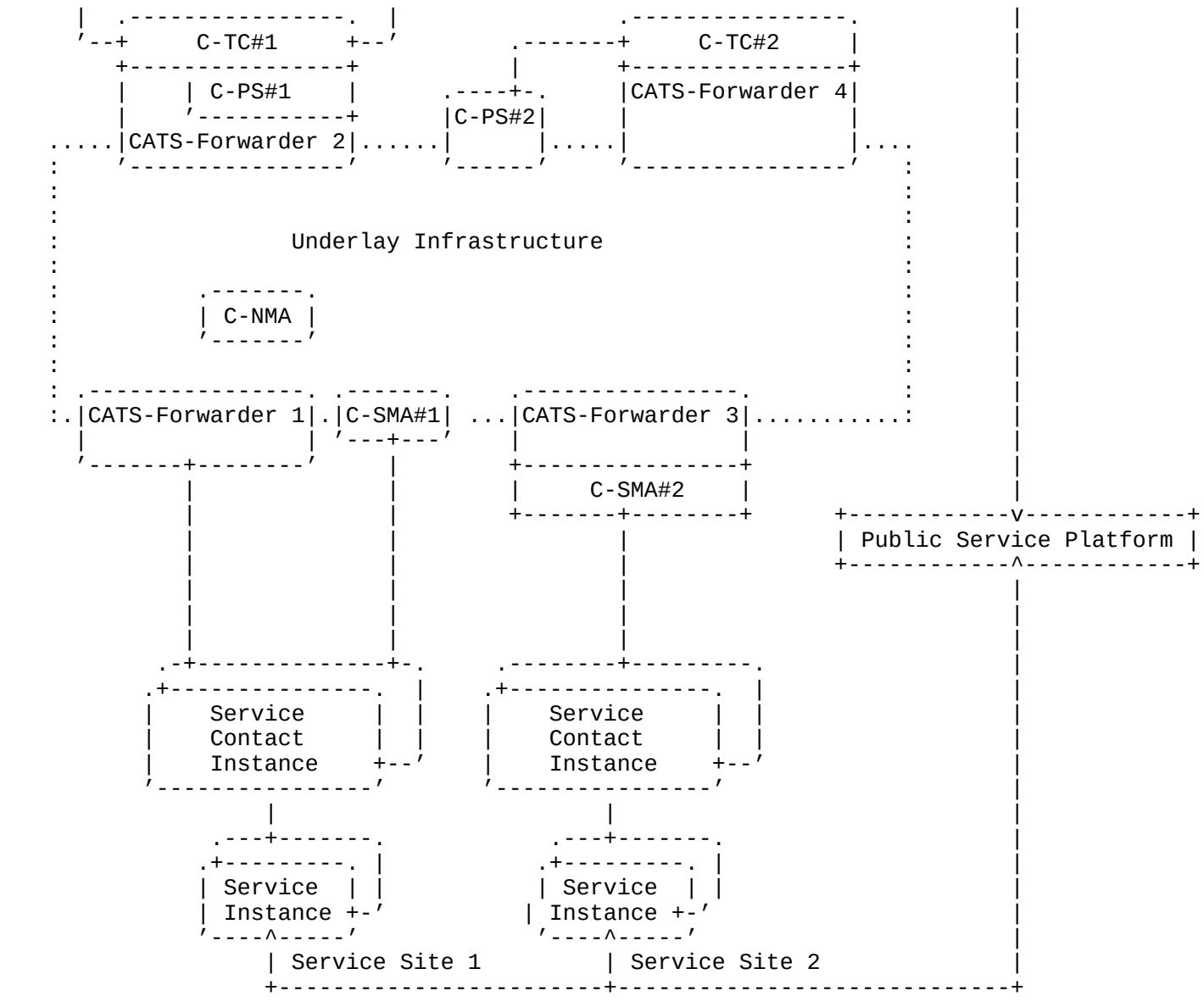
In the CATS framework, the C-SMA collects computing-related capabilities and metrics, and associates them with a CS-ID that identifies the service. The C-SMA then advertises CS-IDs along with metrics to related C-PSes in the network. Computing metrics are numerous and highly variable, which makes them unsuitable for direct dissemination on the network. [I-D.ietf-cats-metric-definition-08] proposes to use normalized metrics in CATS. [I-D.zhangb-cats-service-metrics-op-01] further defines service-oriented metrics and operational procedures for exposing actionable service capacity to CATS control-plane components. This document focuses on the public service platform that provides the catalogue and service context used by those metrics and procedures.

Computing resources are inherently heterogeneous, spanning CPUs, GPUs, FPGAs, ASICs, and other accelerators, each with distinct performance characteristics. This diversity makes it difficult to define a single measurement or normalization scheme that is meaningful across all service providers and hardware types. Normalized scores can also hide service-specific information that is needed when a client requests a concrete service capability.

This document describes a public service platform for CATS. A CS-ID identifies a service, but a CS-ID alone does not tell a client what function the service provides, what input data is required, or how a request should be constructed. Clients need a catalogue that explains the service associated with the identifier. Service sites and other service publishers also need a common place to publish service entries so that clients and other sites can discover, download, deploy, and use those services. The public service platform provides this catalogue and the reference service context used by service-oriented metrics. This document does not redefine the metric semantics or operational procedures specified in [I-D.zhangb-cats-service-metrics-op-01].

Figure 1 extends the CATS functional components from the CATS framework with a public service platform. The platform is a catalogue and publication point: clients query it to generate service requests, and service sites query it to select and deploy services. Both clients and service sites can publish service entries and deployment-related information to the platform.





Client <-> Public Service Platform:
query the platform to generate service requests;
publish service entries and deployment-related information to the platform.

Service Site <-> Public Service Platform:
query the platform to select and deploy services;
publish service entries and deployment-related information to the platform.

Figure 1: CATS Functional Components with Public Service Platform

2. Terminology

This document makes use of the terms defined in [I-D.ietf-cats-framework-24] and the service-metric concepts defined in [I-D.zhangb-cats-service-metrics-op-01]. It also makes use of the following terms:

* Public service platform: A catalogue and publication component that maintains public service descriptions, identifiers, and deployment context for CATS.

3. Public Service Platform

The public service platform hosts the public service catalogue for the CATS framework and serves as a bridge among clients, service sites, and CATS control-plane components. Service sites can discover, deploy, and publish services through the platform, while clients can formulate service requests using stable public service identifiers. The platform binds each service to its input description, deployment requirements, and the service context needed to interpret service-oriented metrics. Service sites can then allocate local resources according to the service units associated with a selected service and report service information and metrics to CATS control-plane components as defined in [I-D.zhangb-cats-service-metrics-op-01]. Table 1 illustrates a typical public service table: an openly searchable and browsable registry for both clients and service sites.

Table 1: Example Public Service Table

ID	Name	Input	Desc	Code	Comp	Stor	Time	GAS	Soft	Pub	Upd	Pop
AR1	AR/VR	Motion capture, voice tracking, eye tracking, environmental sensing.	Receives sensor input and generates AR/VR scenes.	Code link	Multi-thread CPUs, min. 2.0 GHz; GPU higher than RTX 4060.	16 GB DRAM; 256 GB SSD.	<= 1 ms	1	Unity, Unreal Engine	Service Site 1	2026-05	32

|LLM1|LLM |Prompt, |Text |Code|GPU |Model |50 ms |500|CUDA, |Platfor|2026-|128|

	inferen ce	context, generati on paramete rs.	generatio n or question- answering service.	link	cluster or acceler ator pool.	storage and KV-cach e memory.	- 2 s		inferen ce runtime	m Operato r 1	05	
TR1	Model trainin g	Training data, model configur ation, paramete rs.	Training or fine-tuni ng task; returns model artifacts .	Code link	Dedicat ed GPU or acceler ator resourc es.	Dataset storage and checkpo int storage .	Minute s to hours	1	PyTorch , CUDA/cu DNN	Client 1	2026- 05	16
TP1	Intelli gent transpo rtation	Transpor t standard data, traffic informat ion.	Driving or transport ation environme nt sensing.	Code link	CPU >= 4.0 GHz; GPU >= 200 TOPS.	64 GB DDR5 DRAM; 1 TB NVMe SSD.	<= 20 ms	200	Apollo, CUDA	Third Party 1	2026- 05	45
ST1	Simulta neous interpr etation	Speech input and optional interact ion input.	Real-time captionin g or conferenc e translati on.	Code link	CPU >= 3.5 GHz, 16 threads ; RTX 4090-cl ass GPU.	32 GB DDR5 DRAM; 1 TB NVMe SSD; 16 GB GPU memory.	<= 1 s	1	CUDA/cu DNN, Apache Kafka	Service Site 2	2026- 05	21

The service ID identifies a service capability described by the service table. A client can query the service table to find a service of interest, understand the service function and input data format, build a service request using the service ID, and send the request to its ingress CATS-Forwarder. A service site can query the table, find services of interest, download or obtain the associated service code, allocate resources based on the computing and storage requirements, and deploy these services as service instances. A service contact instance can be associated with a deployed service instance that provides the selected service.

The catalogue is not limited to services initially published by service sites. A service entry can be published by a service site, a platform operator, a third party, or a client proposing a new service requirement. Such an entry can describe the requested service

function, input data format, expected service performance, deployment requirements, and reference execution context. After the public service platform accepts the entry according to its publication policy, service sites can query the entry, decide whether to deploy it, and later report the corresponding service contact instances and service-oriented metrics through the C-SMA.

The Computing Requirement and Storage Requirement fields describe the minimum recommended resources for deploying the service. If a service site cannot satisfy these requirements, it is not recommended to deploy the service. A service site can deploy the service with resources greater than or equal to the listed requirements according to local policy and capacity.

The Reference GAS field is a catalogue-level indication of the number of concurrent clients that the published service is expected to support under the listed reference resource configuration. This value is useful when another service site wants to deploy the same service without repeating the initial sizing exercise. If the resource allocated to a service instance just meets the listed requirements, the initial operational GAS can use the Reference GAS as a starting value. If more resources are allocated, the operational GAS needs to be evaluated and is generally larger than the Reference GAS. The operational GAS, as defined in [I-D.zhangb-cats-service-metrics-op-01], is evaluated and reported by each service site through the C-SMA after deployment.

Reference GAS depends on the service type. A training or fine-tuning service normally has Reference GAS equal to 1, because one training job usually consumes a dedicated resource pool for one user or task. An inference service can have Reference GAS greater than 1 when one deployed service can serve multiple users concurrently, such as an LLM inference service. Some low-latency or application-specific inference services may still use Reference GAS equal to 1. The reference computing time listed in the public service table is also a catalogue value measured when the service processes a basic data sample. Operational Computing Time is measured and reported by service sites according to [I-D.zhangb-cats-service-metrics-op-01].

Publisher and Publication & Update Time identify the source and freshness of a service entry. Popularity is a numeric catalogue value that reflects how many times a service is downloaded and deployed by service sites. It can help service sites decide whether a service is worth deploying. Popularity is catalogue metadata and is not treated as a CATS routing metric in this document.

4. Service Modelling with the Public Service Platform

The public service platform supports two basic uses. First, a client can query the platform to understand a service before constructing a request. The client can use the Service ID, Service Name, Description, and Reference Computing Time to determine whether the service matches its needs and to form a service requirement. The resulting request contains the CS-ID and may include additional constraints such as expected service time. The request is then sent to the ingress CATS-Forwarder; candidate selection and forwarding are performed according to the CATS framework and [I-D.zhangb-cats-service-metrics-op-01].

After the client selects a service and a data path to a service site is established, the client uses the Input description to construct the service data sent to the selected service site. The service site processes the data according to the deployed service and returns the result to the client.

Second, a service site can browse the platform, select services it intends to host, and deploy the corresponding service instances locally. The platform provides the service identifier, input description, deployment requirements, code location, and reference service context used for this deployment decision. A service site may follow the reference resource configuration, or it may allocate more resources according to local policy and capacity.

The reference values in the platform help the service site estimate its initial deployment scale. For example, if a service entry has Reference GAS 200 under the reference resource configuration, three equivalent deployments can provide an initial aggregate capacity of about 600 concurrent clients, while four equivalent deployments can provide about 800. If a site allocates more resources than the reference configuration and verifies the result through local testing, it may report a higher operational GAS, such as 260 instead of the reference value 200. The same principle applies to operational Computing Time, which may differ from the catalogue reference value after local deployment.

After deployment, the service site determines the service contact instances and reports the corresponding service identifiers and service-oriented metrics through its C-SMA. These reports form the Computing Service Table defined in [I-D.zhangb-cats-service-metrics-op-01]. This document uses that mechanism by reference and does not define metric encoding, update policy, or selection procedures. In this way, the public service platform supplies the service context, while the service-metric draft defines the metric behaviour and operation.

5. Security Considerations

The public service platform provides catalogue information that clients and service sites rely on for service discovery and deployment context. Implementations should protect the integrity and authenticity of service entries, apply appropriate access control to publication and update operations, and consider the availability of the platform because it may affect service discovery and deployment decisions.

Detailed service descriptions and deployment requirements may expose operational or business information. Operators should control what information is published in the catalogue according to local policy. These considerations are complementary to those discussed in [I-D.ietf-cats-framework-24] and [I-D.zhangb-cats-service-metrics-op-01]. This document does not define specific security mechanisms.

6. IANA Considerations

This document has no IANA actions at this time.

7. References

7.1. Informative References

[I-D.ietf-cats-metric-definition-08]
Yao, K., Li, C., Contreras, L. M., Ros-Giralt, J., and G. Zeng, "CATS Metrics Definition", 15 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-metric-definition-08>.

[I-D.zhangb-cats-service-metrics-op-01]
Zhang, B., Dai, Y., Du, Z., and C. Miao, "Computing Service Metric Definitions and Operation under CATS", 13 May 2026, <https://datatracker.ietf.org/doc/html/draft-zhangb-cats-service-metrics-op-01>.

[I-D.ietf-cats-framework-24]
Li, C., Du, Z., Boucadair, M., Contreras, L. M., and J. Drake, "A Framework for Computing-Aware Traffic Steering (CATS)", 2 April 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-24>.

Authors' Addresses

Bin Zhang (editor)
Pengcheng Laboratory
Sibilong Street
Shenzhen
518055
China
Email: zhangb@pcl.ac.cn

Yina Dai (editor)
Sun Yat-sen University
Sun Yat-sen Street
Guangzhou
510080
China
Email: daiyn5@mail2.sysu.edu.cn

Bowen Shen (editor)
Harbin Institute of Technology
Taoyuan Street
Shenzhen
518055
China
Email: shenbowen@stu.hit.edu.cn

Weizhe Zhang (editor)
Harbin Institute of Technology
Taoyuan Street
Shenzhen
518055
China
Email: wzzhang@hit.edu.cn

Yanchen Qiao (editor)
Pengcheng Laboratory
Sibilong Street
Shenzhen
518055
China
Email: qiaoych@pcl.ac.cn