

Service, Service Instance, & Computing-Aware

IETF#117, July 2023

Mohamed Boucadair (Orange)

Why Are We Having This Discussion?

- Because the Chairs thought there is a confusion among the uses of these terms in CATS documents :-)
- *Echoing definitions* in existing RFCs *might not be accurate* enough for what CATS is committed to do
- Need to make sure we are in *synch* about the intended meaning of key CATS terms, while still allowing to *ease positioning CATS* vs other efforts (SFC, etc.)

RFC 2216

Network Element Service Specification Template

Too restrictive
definition

The definition of a service includes a specification of the functions to be performed by the network element, the information required by the element to perform these functions, and the information made available by the element to other elements of the system. *A service is conceptually implemented within the "service module" contained within the network element.*

NOTE: The above defines a precise meaning for the word "service". Service is a word which has a variety of meanings throughout the networking community; the definition of "service" given here refers specifically to the actions and responses of a single network element such as a router or subnet. This contrasts with the more end-to-end oriented definition of the same word seen in some other networking contexts.

RFC 3198

Terminology for Policy-Based Management

Network-centric

\$ service

(P) The behavior or functionality provided by a network, network element or host [DMTF, RFC2216]. Quoting from RFC 2216 [RFC2216], in order to completely specify a "service", one must define the "functions to be performed ..., the information required ... to perform these functions, and the information made available by the element to other elements of the system". Policy can be used to configure a "service" in a network or on a network element/host, invoke its functionality, and/or coordinate services in an interdomain or end-to-end environment.

RFC 7665

Service Function Chaining (SFC) Architecture

Good but still a
bit network-
centric

Network Service: An offering provided by an operator that is delivered using one or more service functions. This may also be referred to as a "composite service". **The term "service" is used to denote a "network service" in the context of this document.**

Note: Beyond this document, the term "service" is overloaded with varying definitions. For example, to some a service is an offering composed of several elements within the operator's network, whereas for others a service, or more specifically a network service, is a discrete element such as a "firewall". Traditionally, such services (in the latter sense) host a set of service functions and have a network locator where the service is hosted.

Service: A Generic Definition

- 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*, e.g., these resources may be:
 - Exposed by one or multiple monolithic processes (a.k.a. Service Functions (SFs))
 - 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 (e.g., SFC)

Computing Service

- An offering that is made available by a provider by *orchestrating a set of computing resources* (without networking resources)
- Which and how these resources are solicited is part of *the service logic which is internal to the provider*, e.g., these resources may be:

CATS should accommodate all these deployment schemes

- Exposed by one or multiple monolithic processes (a.k.a. Service Functions (SFs))
- 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 (e.g., SFC). How the chaining is implemented is beyond the scope of CATS.

For
discussion

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
- A client may or may not be aware that there are many running instances of the same service

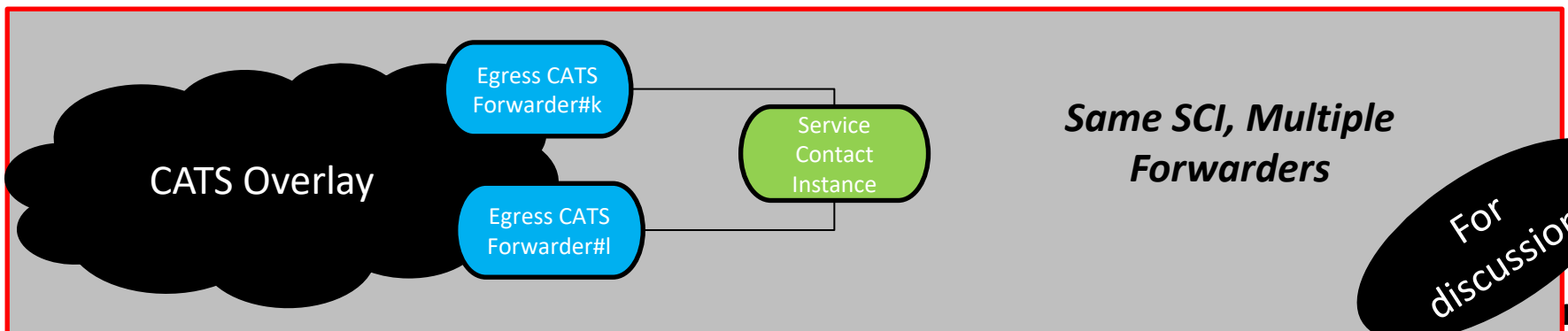
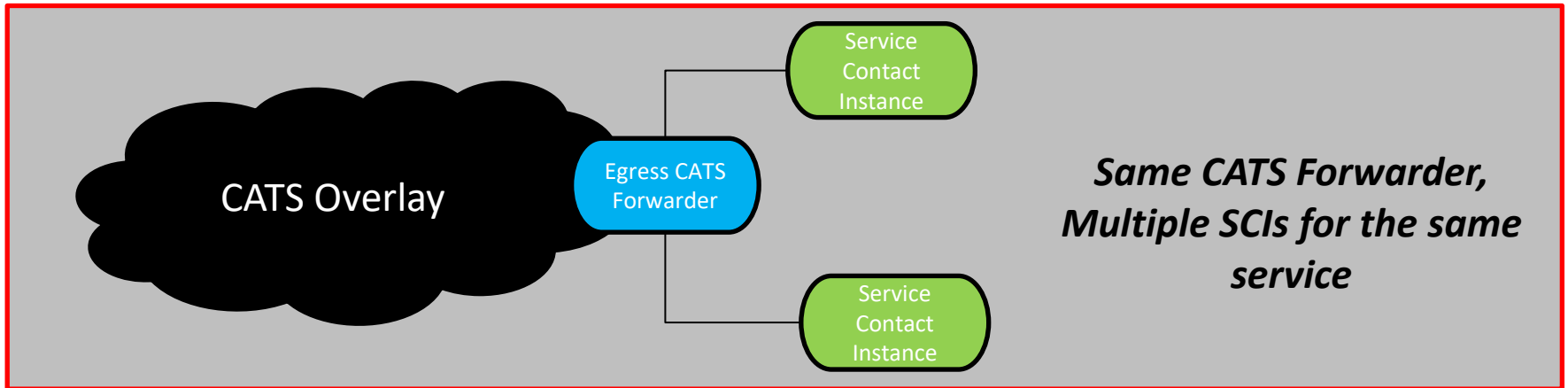
Service Contact Instance

- A *client-facing service function instance* that is responsible for receiving requests in the context of a given service
 - A service request is processed according to the service logic (e.g., handle locally or solicit backend resources)
 - Steering beyond the service contact instance is *hidden to clients*
- A service can be accessed via *multiple service contact instances* running at the same or different locations (service sites)
- The same service contact instance may *dispatch service requests to one or more service instances* (e.g., an instance that behaves as a service load-balancer)

Service Contact Instance

- A *client-facing service function instance* that is responsible for receiving requests in the context of a given service
 - A service request is processed according to the service logic (e.g., handle locally or solicit backend resources)
 - Steering beyond the service contact instance is *hidden to clients and CATS components*
 - In CATS, a service contact instance is reachable via *at least one Egress CATS Forwarder*
- A service can be accessed via *multiple service contact instances* running at the same or different locations (service sites)
- The same service contact instance may *dispatch service requests to one or more service instances* (e.g., an instance that behaves as a service load-balancer)

Sample Service Contact Instance Deployments



For discussion

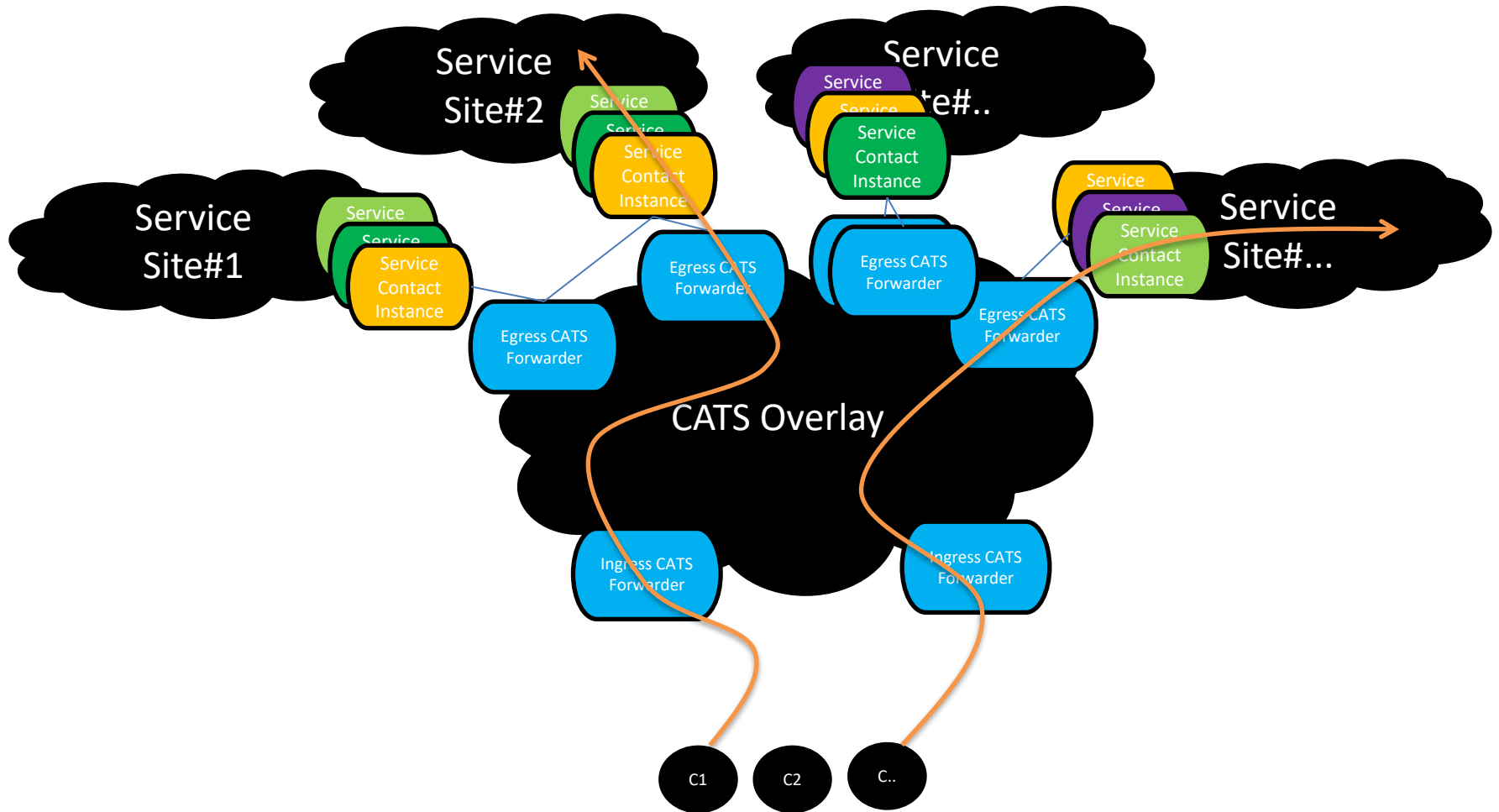
Computing-Aware *

- A * scheme which takes as input a set of metrics that reflect the capabilities/state of computing resources
 - with “*” may be
 - Forwarding
 - Steering
 - Path Computation
 - ...

Steering in CATS

- Steering in CATS is about *selecting the appropriate Service Contact Instance* that will service a request according to a set of network and computing metrics
 - That selection *may not necessarily reveal the actual service instance* that will be invoked, e.g., in hierarchical/recursive contexts
 - *The metrics are aggerated*; that is, they reflect the collective resources involved in a service instance

Steering in CATS: A Realization Example



Summary

- This is not an attempt to be exhaustive, purist, or perform an RFC archeology about these terms, but agree on acceptable grounds for CATS
- Next Steps
 - Ensure draft-ldbc-cats-framework record the outcome of the discussion
 - Same definitions to be used by other CATS I-Ds, typically by referring to the framework I-D