



CAMARA

THE TELCO GLOBAL API ALLIANCE

The need for Runtime Restrictions

January, 2024

Challenge



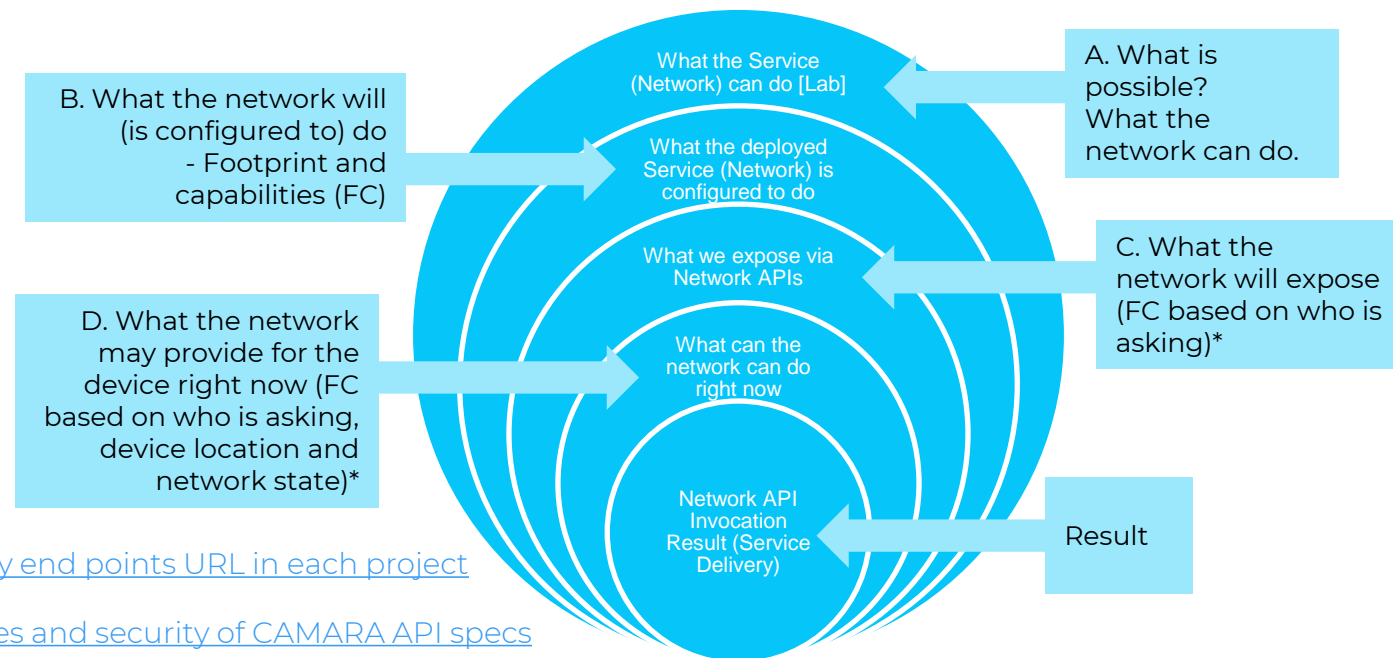
Several subprojects have open issues signaling the need to communicate operator implementation & deployment limits. For example:

- Optional parameters imposed by an API Producer (Operator)
- Supported operations and features at time of API invocation

Reference issues:

- [\[Commonalities->Release Management\]](#) Define Mandatory end points URL in each project
- [\[Identity and Consent Management\]](#) Align securitySchemes and security of CAMARA API specs [\[Closed\]](#)
- [\[DeviceStatus\]](#) Support with Identifier [\[now Discussion 48\]](#)
- [\[SimSwap\]](#) Remove MSISDN from request body [\[Closed\]](#)
- [\[QualityOnDemand\]](#) Add support for DSCP for QoD sessions

Layers of Capabilities



Example Scenario



1. QualityOnDemand API CreateSession Operation has required attribute 'device'
2. Device properties must be at least one of the following
 - PhoneNumber
 - NetworkAccessIdentifier
 - DeviceIpv4Addr
 - Ipv6Address
3. While there is a minimum requirement of 1 property to be provided, schema doesn't describe which one(s) maybe supported by an operator
4. Operator / Aggregator (API Producer) may only support subset of these options
 - i.e. PhoneNumber, Ipv6Address

There is currently no mechanism to relay to API consumers which optional capabilities/features/parameters are supported and enabled by API Producers and API Consumers!

Options are either for operators to modify Service API, or identify least common denominator, both of which defeat goal of CAMARA!

Scenarios that can benefit



CAMARA
THE TELCO GLOBAL API ALLIANCE

1. Supported Properties (e.g. Device Identifier)
2. Operations Supported / Enabled per Operator, Device, Policy, Subscription, Network State, etc.
3. Regulatory limitations (e.g. Increased Minimum Radius, reduced Maximum Radius)
4. Supported security schemes per flow type
5. Disabled capabilities due to Location and/or Network status



[IETF RFC 8008](#) – Content Delivery Network Interconnection (CDNI) Request Routing: Footprint and Capabilities Semantics

[3GPP TS 29.500](#) – Extensibility Mechanisms

[TMF 630 - REST API Design Guidelines](#) – Part 2 Describes Extension Patterns

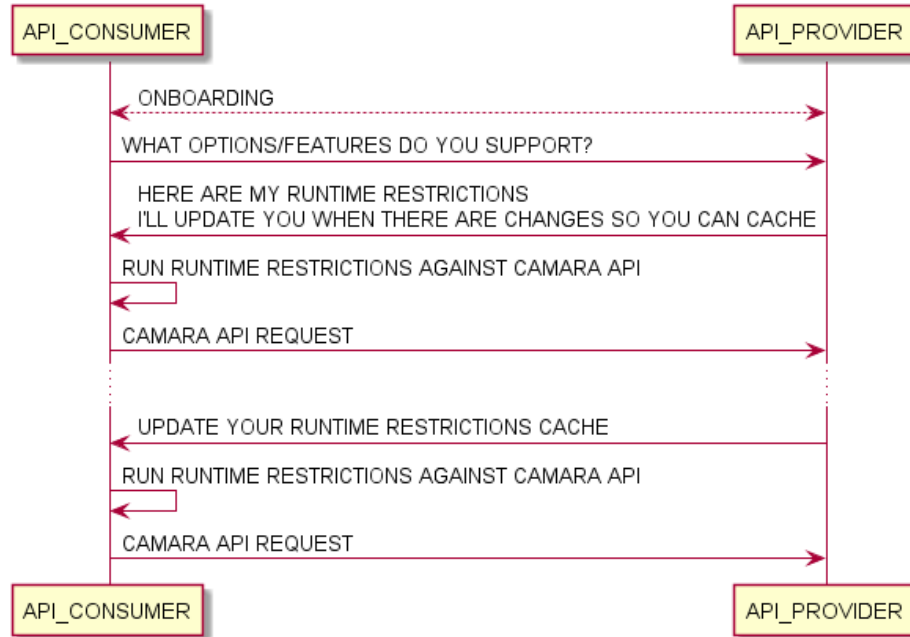
Ground Rules and Proposal



A CAMARA API processed against an API provider's Runtime Restrictions MUST result in valid requests for the API and associated schema

The Runtime Restrictions interface MUST never pass limitations that result in invalid API request.

1. Any instance may have metadata (deprecated, readOnly) or general (type, enum, const) restrictions.
2. Use JSON Schema Validation for defining 'Restrictions' and apply to primitive types "numeric", "strings", "array", "object [instance, excluded properties]"
3. Parameters may be restricted as follows:
 - Deprecate use of a parameter. This can only be done if the original schema does not have 'required: true' for this parameter specification.
 - Mandate use of an optional parameter. This can only be done if the original schema does not have 'deprecated: true' for this parameter specification.
 - Allow empty values. This can only be TRUE if the schema did not explicitly specify its value as false.
4. Restrictions may apply to subschemas that are applied with logic keywords "anyOf" or "oneOf"
5. Setting a referenced item to 'readOnly', effectively noting that its value will be ignored by the server
6. If the API producer doesn't implement all operations defined in CAMARA API in the respective version, such operations can be restricted by either declaring the operation
 - 'deprecated', as long as the original schema did not explicitly specify its value as false,
 - 'not available', for temporary unavailability
 - or 'not implemented', if the operation hasn't been implemented.



Example Structure

```
▼ [Restrictions ▼ {
description: The restrictions and scope (operations, parameters, operation request bodies or entire schema) a set of
restrictions applies to.
cacheId > [...]
Identifiers > [...]
version > [...]
context > [...]
subjects* > [...]
restrictions*
  ▼ {
  minItems: 1
  A list of restrictions that limit what can be passed to an API. when applied, it MUST result in a valid request
  per the original schema specification.
  ▼ {
    oneOf ->
      AnyInstanceMetaRestriction > (...)
      AnyInstanceRestriction > (...)
      NumericInstanceRestriction > (...)
      StringInstanceRestriction > (...)
      ArrayInstanceRestriction > (...)
      ObjectInstanceRestriction > (...)
      ParameterRestriction > (...)
      OperationRestriction > (...)
  }
  }
}
```

Additional Work is needed!

While 'Runtime Restrictions' provide a programmable way to identify restrictions applied by an API provider, it lacks the ability to group these restrictions for quickly turning on/off and associate with footprints.

Examples



CAMARA
THE TELCO GLOBAL API ALLIANCE

```
{
  "subjects": [
    "https://github.com/camaraproject/QualityOnDemand/blob/main/code/API_definitions/qod-api.yaml#/paths/~1sessions/post/requestBody/content/application~1json/schema/CreateSession/properties/device/"
  ],
  "restrictions": [
    {
      "excludedProperties_rr": [
        "https://github.com/camaraproject/QualityOnDemand/blob/main/code/API_definitions/qod-api.yaml#/components/schemas/Device/ipv4Address",
        "https://github.com/camaraproject/QualityOnDemand/blob/main/code/API_definitions/qod-api.yaml#/components/schemas/Device/networkAccessIdentifier"
      ],
      "rrType": "ObjectInstanceRestriction"
    }
  ]
}
```

*ipv4Addresses and
NAIs are not
supported!*

Scenario 1

```
{
  "subjects": [
    "https://github.com/camaraproject/DeviceLocation/blob/main/code/API_definitions/location-verification.yaml#/components/schemas/Circle/radius"
  ],
  "restrictions": [
    {
      "minimum_rr": 3000,
      "maximum_rr": 30000,
      "rrType": "NumericInstanceRestriction"
    }
  ]
}
```

*Minimum radius
supported is 3000
[meters] vs. 2000*

Scenario 2

API Submission Template (DRAFT)



CAMARA
THE TELCO GLOBAL API ALLIANCE

Field	Description
API family name	Capability and Runtime Restrictions
API family owner	T-Mobile US
API summary	<p>CAMARA Service APIs are designed with many optional parameters and features. It's unreasonable to expect each API Producer (i.e. Operator) to support all these optional parameters. In addition, some supported features and parameters may not be enabled at Service API invocation time, based on network state, who might be invoking and/or for whom/which device, location, ...) There is currently no mechanism to exchange such information with API Consumers (i.e. Application Service Providers (ASP)/Developers/Aggregators) and keep the CAMARA APIs the same across the Exposure Gateways.</p> <p>API Family is intended to cover the following areas:</p> <ol style="list-style-type: none">1. Exchange of runtime restrictions (i.e. not supported parameters/features)2. Exchange of capabilities (i.e. enabled/not enabled a set of parameters/features)3. Topology exchange (i.e. abstraction) for capability-footprint association <p>For the first area following examples can be given:</p> <ol style="list-style-type: none">1. Device identifier in QoD can be of type Phone Number, IPv4 Address, IPv6 Address, Network Access Identifier (NAI). One operator may support all of these identifiers in which case there will not be a need to list any restrictions towards the schema in the QoD API, however another operator supports only Phone Number, thus will need to inform ASPs not to use them.2. If there is a regulation specific maximum accuracy level that must be set greater than the minimum Radius defined in 'location-retrieval.yaml', operator must be able to overwrite this new minimum. <p>For the capabilities following example can be given:</p> <ol style="list-style-type: none">1. While the operator may not support only one of the default set of QoD Profiles, at the time of invocation, one or more of the supported QoD profiles may not be available due to network state and/or location. Invocation of QoD with not-enabled QoD profile will result in error and lead to bad developer experience. Operators must be able to efficiently exchange this information.
Technical viability	Yes (reuse of JSON Validation Schema with little to no impact to current API designs)
Commercial viability	This is not a product, but rather Service Management API which falls under CAMARA purview.
YAML code available?	Yes (for runtime restrictions)
Validated in lab/productive environments?	In progress
Validated with real customers?	No
Validated with operators?	No
Supporters in API Backlog Working Group	



CAMARA

THE TELCO GLOBAL API ALLIANCE