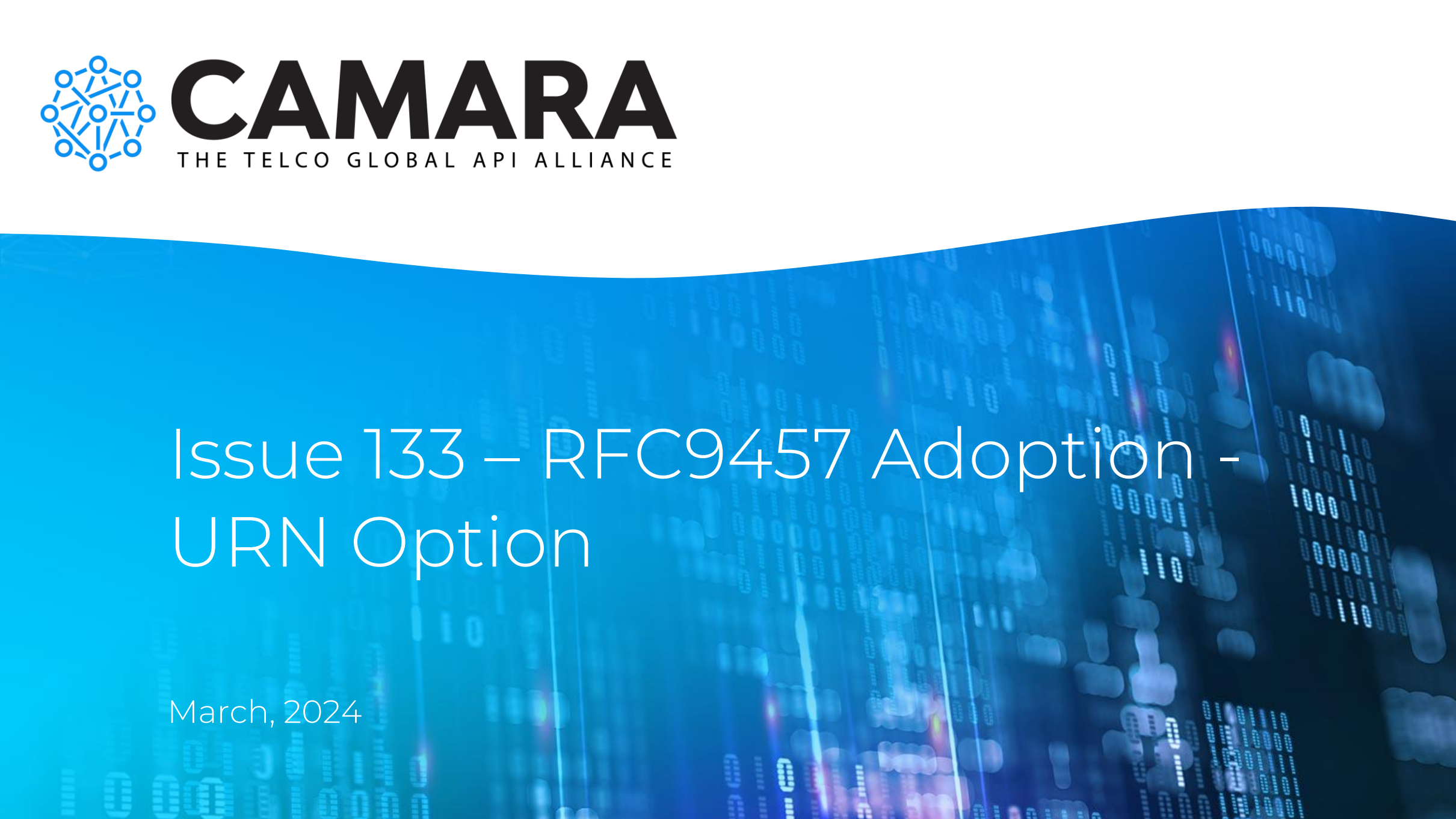

Issue 133 – RFC9457 Adoption - URN Option

March, 2024

Why adopt RFC 9457?

CAMARA
THE TELCO GLOBAL API ALLIANCE

- We need a future proof, extensible schema for problem details (beyond pure HTTP status codes)
- There is nothing out there in the industry, but RFC 9457 is promising and already adopted by some organizations
- RFC 9457 would allow the use of HTTP Standard, CAMARA specific and even API specific problem types side-by-side in a IETF standardized way
- With RFC 9457 CAMARA wouldn’t add another proprietary error format for developers to deal with

Other formats and RFC 9457 (referenced in Issue 31)

CAMARA
THE TELCO GLOBAL API ALLIANCE

No clear direction from others

RFC 9457 Properties / Company	Link	type (URI)	*status* - the HTTP status code	*title* - a short, human-readable summary of the problem type	detail - a human-readable explanation specific to this occurrence of the problem.	*instance* - string containing a URI reference that identifies the specific occurrence of the problem
Google	https://cloud.google.com/apis/design/errors	proto3 - derived during payload decode http - details (struct) property's "@type" property	proto3 - code property (follow http values but they use their own codes) http mapping - code property	proto3 - no equivalent in header http mapping - status property	message property for proto3 and http	
Meta	https://developers.facebook.com/docs/whatsapp/cloud-api/support/error-codes/#error-codes	type property (string)	code property (they use their own codes)	part of their message property which is "Combination of the error code and its title."	details property	messaging_product and fbtrace_id
Amazon EC2	https://docs.aws.amazon.com/AmazonS3/latest/API/ErrorResponses.html, https://docs.aws.amazon.com/AWSEC2/latest/APIReference/errors-overview.html		code as a string that is given subcodes using a '.' notation		message property	
Azure	https://learn.microsoft.com/en-us/rest/api/storageservices/status-and-error-codes2	Error Code (String)	recommends using the http status code (not in the body)	Error Code (String)	message property	additional information that varies and *may* be provided
Azure - Table Storage (Odata Spec v 4 Section 19)	https://learn.microsoft.com/en-us/rest/api/storageservices/status-and-error-codes2		code (it is a substatus of the http error code)	message property	details sub structure	target
Analysis >>>>>>		Inconsistent when present but format is basically a string (url or just a string); otherwise derived during decoding (Google proto3)	Code property name is consistent but format and values are not. Some note that if the HTTP status code is sufficient, use it.	No consistent name or presence. String or struct data type is used. Values vary (as expected)	Present as a struct or string and either called message or details.	Present occasionally but some services respond back with product and a trace (hiding instance info indirectly)

Issue 133 – Proposal Basis

The type property definition (URI) in the RFC is too broad, becoming problematic without restriction. This is compounded by RFC 9457:

- encouraging use of documentation when the URI is a locator (URL or URN that location services can use).
- noting risks of relative URIs without strong guidance

Some *implementations* resolved this in different ways, restricting property values, while remaining compliant.

From https://opensource.zalando.com/restful-api-guidelines/#176

> "**Note**: Problem type and instance identifiers in our APIs are not meant to be resolved. RFC 9457 encourages that problem types are URI references that point to human-readable documentation, but we deliberately decided against that."

This does not resolve Relative URI concerns.

From https://www.belgif.be/specification/rest/api-guide/#error-handling

> "Note that using href instead of type for documentation intentionally deviates from the recommendation in the RFC. href allows use of a URL for documentation purposes that may change over time, while type can be specified as a URN that must remain stable. This is especially useful for API-specific problem types for which the documentation URL may depend on technical aspects, like deployment environment."

This resolves both issues above and is the basis for this proposal.

type property usage recommendations

CAMARA
THE TELCO GLOBAL API ALLIANCE

Recommendation A - The **type** property is *restricted* to URNs as defined in RFC 8141 from a managed URN namespace managed by CAMARA. These URNs will not be relative nor are they intended to resolve to locators.

This creates a persistent, location-independent resource identifier. It avoids issues raised by the RFC and others and is RFC compliant. It does not follow the RFC's encouragement of using locators and resolving to documentation.

Recommendation B - Developer targeted information generally describing the error or related documentation will be a URL contained in a **href** (or other name*) property in the error.

This deviates from the RFC encouragement but maintains compliance. The property can be dropped for production systems.

* - TMF had (a still open) issue https://projects.tmforum.org/jira/browse/AP-2628 created in March 2021, which wasn't considered for the v5 of TMF 630 (the Guidelines) and now to be closed. They compared the TMF and RFC formats, showing a parameter "referenceError" which is "URI of problem type documentation"

URN Namespace Management

Proposed format: *urn:camara:<..>*

The type property's value is interpreted as follows:

- No type, aka "about:blank" – this type is for simple errors where no additional information needs to be provided. This information might not come from a CAMARA API but other servers in the application stack or path
  - HTTP status code only, without any additional semantic
  - If Title is provided, it must be the one defined for the HTTP status code, no deviations.
  - Status property may be mandatory
- CAMARA wide defined problem types: "type" = "urn:camara:common.errors…"
  - A fixed title
  - The HTTP status code for it to be used with
  - May have problem type specific extensions
- CAMARA API specific problem types: "type" = "urn:camara:<api-name>.errors…"
  - A (fixed) title (which does not need to have the API name in it again)
  - The HTTP status code for it to be used with
  - May have problem type specific extensions

For further discussion:

- addition of api specific versions in the URN (does not make sense but is possible). Cloud Event type recommends this [link]
- sub-typing via '/', e.g., "/<code>/<sub-code>", "urn:camara:common/invalid_device/ipv4addressNotFound"
- vendor or operator specific sections via '/', e.g., "urn:camara:common/invalid_device/vendorspecific/operatorA/…"
- Unify CAMARA events and errors under common management in the CloudEvent type, i.e., "urn:camara:<api-name>.events…" when the type is a CAMARA defined. Otherwise follow the Cloud Events recommendations. Per specification, the property is "a value describing the type of event related to the originating occurrence" [link] but the type is a string whose value "SHOULD be prefixed with a reverse-DNS name. The prefixed domain dictates the organization which defines the semantics of this event type." [link] this.

Other impacts to the existing base (outside of the type property)

CAMARA
THE TELCO GLOBAL API ALLIANCE

As originally identified at the top of the Issue thread

- Rename the current code fields to title, with further restriction that its value is the phrase defined by the HTTP Status code then the type property is not present, i.e., “about:blank”
- Rename the message field to detail, with no other change to how that field is currently used
- No change to the status field

Other changes that need to be incorporated as part of a PR.

1. Add an enum (string), CAMARACommonErrors, to CAMARA_Common.yaml. Its purpose is to contain the URNs of Error types that span many CAMARA APIs (sub-projects).

2. Add the following statements to the Design Guidelines

- CAMARA wide (cross API) defined problem types (URNs) MUST be defined in CAMARA_Common.yaml file as enum values in the CAMARACommonErrors string (enum).
- CAMARA API specific problem types (URNs) MUST be defined in their respective API (yaml) file as an enum list and named ‘APISpecificErrors’. Definitions for each code MUST be provided as part of the APISpecificErrors’ description property.
- Per RFC9457, API consumers SHOULD NOT parse the ‘detail’ for more information. This human readable explanation intended to help the client correct the problem.
- All APIs MUST use the Error schema (in its agreed form) defined in the CAMARA_Common.yaml file

Open Issues

CAMARA
THE TELCO GLOBAL API ALLIANCE

Issue 1: Content-Type support: RFC 9457 specifies the Content-Type as “application/problem+json” but it *may* be a client concern.

Do we mandate other Content-Types in our interfaces? If so, what?

Given that this is an issue broader than just errors, should we assume we maintain our current Content-Type and adjust later once we have a stance on Content-Types required to be supported in CAMARA?

TMF had (a still open) issue https://projects.tmforum.org/jira/browse/AP-2628 created in March 2021, which wasn’t considered for the v5 of TMF 630 (the Guidelines). They have a proposal addressing how to enable a migration:

- “The IANA registered media type "application/problem+json" may be used to differentiate from the existing Error defined in TMF630-1 3.4 which specifies application/json. The HTTP Content-Type header of a response may be used to explicitly indicate if the existing or RFC7807 compliant problem details are used.
- The HTTP Accept header can be used to indicate that the client is prepared to accept an application/problem+json body of an RFC7807 response which allows introducing the IETF standard problem details in a backwards compatible way where the clients choose the format.”

CAMARA

THE TELCO GLOBAL API ALLIANCE