**TM Forum Open APIs**

**Conformance Certification**

*Company Name:* ***Totogi***

*TM Forum Open API Name:*

***Resource Catalog Management API TMF634***

*TM Forum Open API Release Version:* ***21.0 / V4.1***

**Report Date: 25th August 2021**

1. **What Product or Solution does your API support?**

The Totogi BSS has encapsulated and instantiated the Resource Catalog API TMF634 enabling standardized mechanisms to manage resource catalog entities.

Communication Service Providers (CSPs) are constantly moulding and adapting their platforms to communicate with each other. Every new service or product that is delivered to the market requires change to a vast number of systems. Launch times lag the market need and revenue opportunities are lost.

Furthermore, the CSPs need to retain customers and excite those customers into buying more services. The range of services is expanding and the CSP has to adopt new content and create a more personalised experience that delights the customer and extends the size of wallet the CSP can target.

As the range of services are extended the CSP is forced to upgrade or change existing systems, processes and channels while adding new, complementary platforms to evolve to a Digital Services Provider (DSP).

To facilitate this move to the DSP model the Totogi product suite leverages the power of the public cloud. Why? Because the Hyperscalers are investing billions in delivering the best development, data management and performance tools. Totogi leverages that investment and delivers webscale performance and availability.

Totogi BSS is the public cloud based platform that enables a CSP to drive its evolution to becoming a DSP. As it evolves it will incorporate all the TMForum APIs and enable the CSP/DSP to deliver new services by building applications in low-code/no-code tools that use the instantiated APIs.

**Data Management:** All our APIs store data into a TMForum compliant data structure. This makes the data available via the public cloud and enables any tool to access information through the API mechanism.

**Analytics and Reporting:** Using Hyperscaler reporting tools the CSP/DSP can rapidly build dashboards and deep analytics on all aspects of their subscribers and their behaviours.

**Public Cloud Architecture:** By having the data on the public cloud you automatically gain the advantages of backup and disaster recovery that are built into the Hyperscaler platforms.

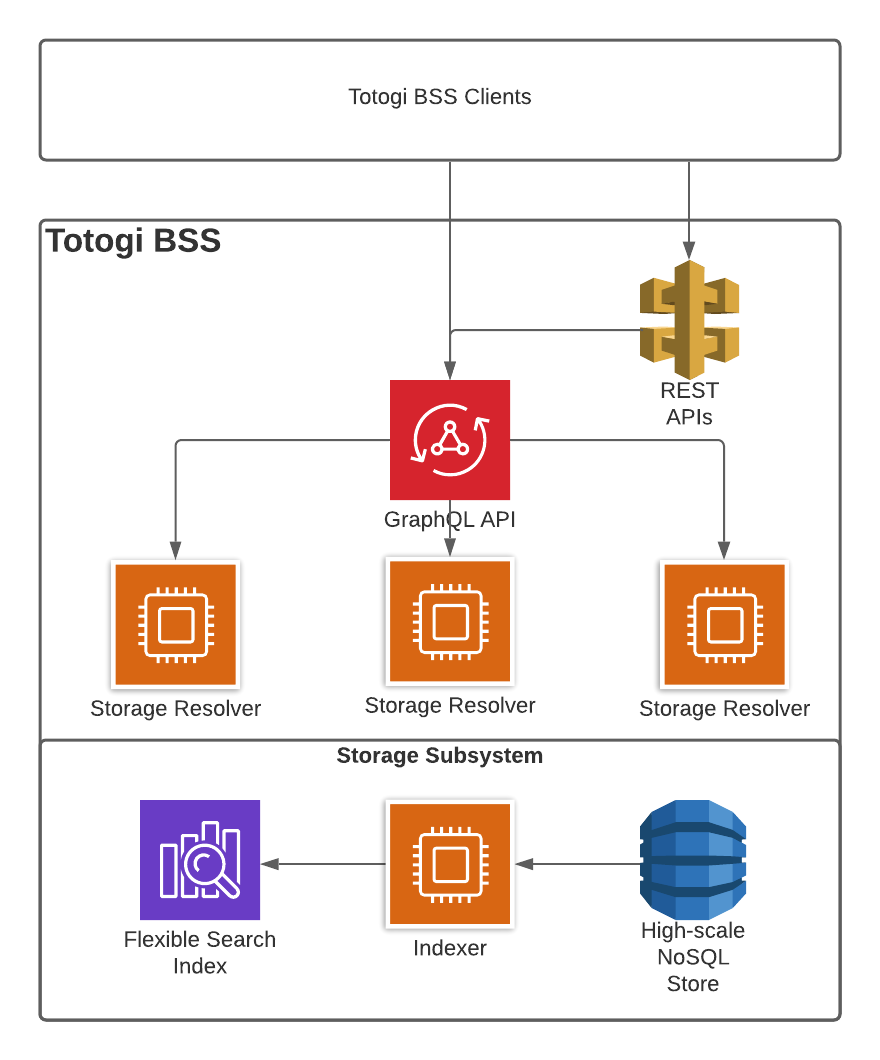
**Rapid Application Development:** The CSP/DSP is now freed from vendor product roadmap lifecycle or internal development backlog that prevents solutions being delivered to meet immediate market need. The Totogi BSS allows the CSP/DSP to solve problems rapidly with low-code/no code tools and writing to open standards.

1. **Overview of Certified API**

The Totogi BSS Resource Catalog Management API is functionality in the core Totogi BSS APIs to manage the entire lifecycle of the Resource Catalog elements and the consultation of resource catalog elements during several other processes such as ordering. It is compliant with the TMF634 specifications and offers API access to creation, update, retrieval and deletion of Resource Catalog elements.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Resource** | **API Overview** | **Operations Supported** |
| 1 | Resource Catalog | The root entity for resource catalog management. A Resource Catalog is a group of Resource Specifications made available through Resource Candidates that an organization provides to the consumers (internal consumers like its employees or B2B customers or B2C customers). | * GET * PATCH * POST * DELETE |
| 2 | Resource Category | Resource Category is used to group Resource Candidates in logical containers. Categories can contain other categories. | * GET * PATCH * POST * DELETE |
| 3 | Resource Candidate | Resource Candidate is an entity that makes a resource specification available to a catalog. A Resource Candidate and its associated resource specification may be published - made visible - in any number of resource catalogs, or in none. | * GET * PATCH * POST * DELETE |
| 4 | Resource Specification | Resource Specification is a base class for representing a generic means for implementing a particular type of Resource. In essence, a Resource Specification defines the common attributes and relationships of a set of related Resources, while Resource defines a specific instance that is based on a particular Resource Specification. | * GET * PATCH * POST * DELETE |
| 5 | Logical Resource Spec | Base class that is used to define the invariant characteristics and behavior (attributes, methods, constraints, and relationships) of a Logical Resource. | * GET * PATCH * POST * DELETE |
| 6 | Physical Resource Spec | Concrete class that is used to define the invariant characteristics and behavior (attributes, methods, constraints, and relationships) of a Physical Resource. | * GET * PATCH * POST * DELETE |

1. **Architectural View**



1. **Test Results**

