tmforum
inform innovate accelerate optimize

Frameworx How-to Guide

Frameworx Open Digital API Business Guide

Open Digital Business Scenarios and Use Cases

TR241
Release 14.5.1
March 2015

Latest Update: Frameworx Release 14.5	**TM Forum Approved**
Version 0.3.1	**IPR Mode: RAND**

Notice

Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

Table of Contents

List of Figures

Executive Summary

The extraordinary successes of 2-sided business model as revenue engine during 2013-2014 have shown how the digital economy has tapped on new areas of revenue outside the traditional value chain. The combined value effect of a “network of networks” has caused global shifts of customer spending patterns. Communication Service providers have had a strong control over the breadth of the complete value chain, where the shift is rapidly moving away from rudimentary “access” towards an “always connected” economy.

The availability of affordable and powerful devices such as tablets or smartphones has lowered the entry barrier to new Digital economy players. Nevertheless, the establishment of an eco-system supporting Digital Services is still a large barrier.

Indeed, digital services partnerships invariably result in systems integration and data sharing between partners and that is achieved through interactions aka Touch Points

To date, efforts of Service Providers, such as CSP or other large Enterprises, in attracting DSP partners and integrating with them have resulted most of the time in manual processes or custom-building for each individual partnership. It is clear that this approach is inefficient, and has resulted in high costs of operation, slow time-to-value, poor customer experience and lack of transparency in partner relationship.

In this context, the TM Forum Community can provide simple Access to Telco Features which can be used in areas far beyond the Telco Industry alone. Instead of focusing on providing standard Telco Services (like Voice API), the Open Digital API provide a set of business features in the prime competence areas of Telco’s like Billing, Inventory, Catalog Management etc.

In addition, the community has learned that independent fragmented allegiances are not sufficient for Digital Business Models which demand Global availability, developers wants quick snap-ins without prior Telco expertise and product lifecycle with extremely short intervals.

So, standardizing partnering methods is a priority and it is imperative to build out the required standards quickly. Being able to proceed in a repeatable and industrial manner based on reusable components with exposed APIs will be key for standardization. Indeed, use of standard management API will facilitate interoperability between partners, reduce integration cost and provide additional benefits such as:

- Lowering the entry barrier to combine Communication & Information with own vertical industries products, spanning national borders and specific brands
- Availability of a simple and modern interface Technology requiring minimum domain expertise and allowing quick setup and deployment
- Reduce administration of multiple external disparate interfaces
- Faster capitalization of assets (increased usage through micro and cross industry exposure).

This document presents the result of the work performed up to now. Its main purpose is to present the value of the TM Forum API to Open Digital Economy players without requiring IT Development skills, formal TM Forum training or knowledge of Communication Service Provider internal ecosystem.

As such, it is written in a business language vocabulary and doesn’t refer neither to Frameworx concepts and vocabulary nor REST specification patterns.

It is structured as follows:

- An overview of Open Digital API’s presenting:
  - The business value they can bring/provide
  - How they can be used in a typical business scenario
- An high level description of each available API
- An illustration of their use, based on description of real world business scenarios
- A description of resources managed by each API.

1. Introduction

This document provides an overview of the TM Forum API and is targeted to a business level audience.

Its main purpose is to present the value of the TM Forum API to Open Digital Economy players without requiring IT Development skills, Formal TM Forum training or knowledge of Communication Service Provider internal ecosystem.

As such, it is written in a business language vocabulary and doesn't refer either to Frameworx concepts and vocabulary or REST specification patterns

It is structured as follows:

- An overview of Open Digital API's presenting:
    - The business value they can bring/provide
    - How they can be used in a typical business scenario
- A high level description of each available API
- An illustration of their use, based on description of real world business scenarios
- A description of generic usage patterns (pending).

1.1. Who is this for?

This document is intended for any party wishing to play a role in the open digital economy – Commercial, Operations and Technical management in CSP, Digital Service Providers, Content Providers and Brokers, software suppliers, Systems Integrators, etc. – who want to use/bundle simple content/communication services traditionally only available to Telco/Cable providers & partners.

It should have optimal benefit for players in industries outside ICT (example Automotive, Health Care ...), or for small technology startups that cannot afford high investments.

The specific focus is on Business people wishing to know what they can achieve by using TM Forum API. In addition, IT Architects and developers will benefit from an overview of available API.

1.2. High level business scenarios /challenges

With the evolution of Digital Services and Communications Service industries there is a growing need to deliver new services by combining the capabilities of multiple organizations.

To reach this goal, use of standard management API will facilitate interoperability between partners, reduce integration cost and provide additional benefits such as:

- For vertical industries
    - Lower the entry barrier to combine Communication & Information with own Products, spanning national borders and specific brands
- For start-ups and SME
    - Availability of a simple and modern interface Technology requiring minimum domain expertise and allowing quick setup and deployment

- For Telco & Information Providers:
    - Reduce administration of multiple external disparate interfaces
    - Faster capitalization of assets (increased usage through micro and cross industry exposure).

1.3. Where did it come from?

This material is based on practical experience of TM Forum member companies involved in API specification within the Open Digital Economy program. It is also based on contributions arising from:

- The ABCs of multi-party digital services catalyst demonstrated at Digital Disruption San Jose 2013
- Use cases provided by TM Forum members.

2. Open Digital API Overview

2.1. Benefits of API

Digital services partnerships invariably result in systems integration and data sharing between partners and that is achieved through interactions aka Touch Points

Figure 1 depicts these interactions in a simple use case based on a mix of real world offers where only a customer and two partners are involved in digital service management It is a typical example .

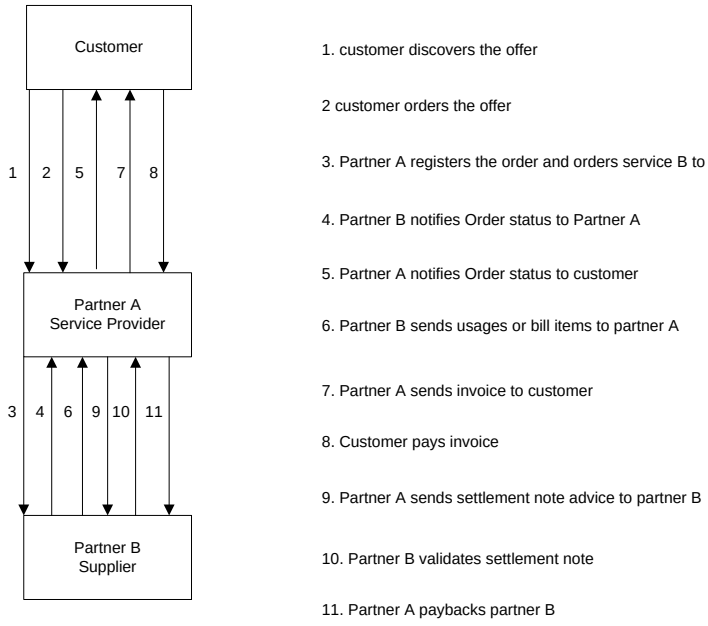

Figure 1 - High level interaction between partners

To date, efforts of Service Providers, such as CSP or other large Enterprises, in attracting DSP partners and integrating with them have resulted most of the time in manual processes or custom-build for each individual partnership. It is clear that this approach is inefficient, and has resulted in:

- high costs of operation,
- slow time-to-value,
- poor customer experience,
- And lack of transparency in partner relationship.

Given that the key added value of the Service Providers is access to a large pool of known customers, an inability to rapidly deliver repeatable service and partnering at industrial scale is a major issue.

So, standardizing partnering methods is a priority and it is imperative to build out the required standards quickly. Being able to proceed in a repeatable and industrial manner based on reusable components with exposed APIs will be key for standardization.

In order to meet these challenges, TM Forum, within the Open Digital Economy program has been working with service providers and suppliers to develop APIs. They enable the open digital ecosystem and provide critical management functionality to digital services that rely on multiple partners and systems operating in a complex value chain.

To date, the Open Digital Economy program has delivered 12 APIs

- Product ordering API
- Catalog management API
- Trouble ticketing API
- Service level agreement API
- Performance management API
- Customer management API
- Party management API
- Usage API
- Billing API
- Product Inventory API
- Service Inventory API
- Resource Inventory API

2.1.1. Which API for which process?

The diagram below describes the different steps of typical customer & related partner journeys and the API that can be used at each stage. It doesn't display all APIs that can be used at a given step but only the ones that are already specified by TM Forum

The reasons why a given API is used at a given step are detailed in the following sections.

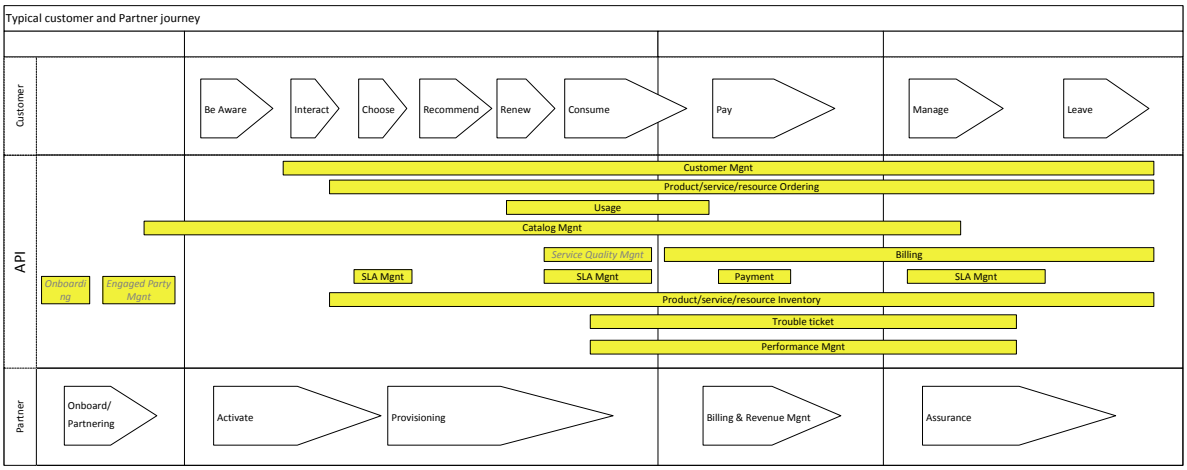

Figure 2 - Use of API in typical customer and partner journeys (sketch)

2.1.2. Building solutions in a value network

One can assume that each API is used equally throughout the relationship duration, however, in reality; its use is (usually) concentrated at different stages. This chapter illustrates the prerequisite to every customer journey in a multifaceted complex relationship environment. The customer (and partner) journey have several interactions (Moments of Truth) at different stages during the lifecycle of their relationship, where the API's have been designed to address a small subset (the minimum –so to say).

The relative placement of the API's or the similarity between partners/customers may also be questioned –

i) it can be argued that catalog management is used also extensively during the consume phase (example upselling/cross –selling) or also along the full cycle, or

ii) That customers may also be on boarded since a while (see crowd-sourcing, ebay ® (direct biz m), Amazon ® (JIT Procure Bizm).

Nonetheless, it is helpful to have a quick glance overview to see which features apply where.

In Figure 3, below, we differentiate quickly the different requirements that are placed on API's used in a traditional value chain, and contrast this to the challenges in a value network.

In contrast to a Local Integration Service (ad-hoc regional value shop) (top of picture), the Value Network (bottom part of the picture) may have very short setup times (Quick start/Quick Die) or cycles – i.e. too short to negotiate a unique contract, or requiring change so constantly, that the Business Case is not too low for operational support by DSP technology department.

The picture is also representative of the increasing challenges of providing more flexible interfaces as time passes. The challenges of the 90's (turnkey solutions) (top part), have been superseded by and increasing number of interfaces (1:1) and complexity to partners (middle part of picture), and the introduction of a cooperative partnership (bottom part), where no single player has absolute dominance.

The Parties to the Open Digital API's (Marked as green stars) cannot assume that the interfaces underlie negotiations of individual partnerships. Since these API's is offered globally, it builds a fundamental cornerstone upon which further feature rich differentiated services can be built upon. Hence, the focus shifts towards supporting newer economy extended service, by lowering the entry barrier globally for existing business services (like catalog management, billing, trouble ticketing etc.)

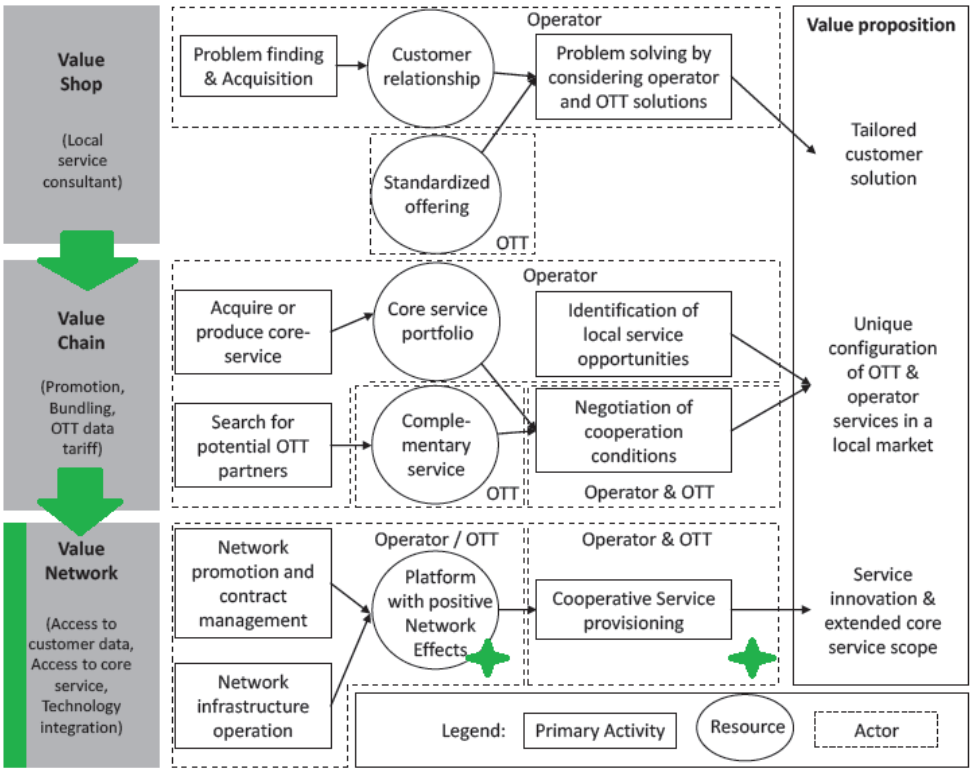

Figure 3: Successive Value creation through -shop, -chain and network (from Cooperative Operator and OTT value creation Patterns –based on Fjeldstad and Haanes; Ostervalder 2004)

As illustrated by Figure 3, the Open Digital API can, of course, be used in a wide variety of scenarios. In addition, they should simplify cooperation in a value network, where the network facilitates dynamic and cooperative behavior beyond what is possible today in a value chain, allowing delivery of services which are not feasible with standard onboarding or partnering processes available today.

2.2. Partner Onboarding: Preliminary steps before API use

Indeed, long before a product (/service) is sold to a customer or made available to a community of developers, an enterprise must establish its business model through partnerships in the value chain and /or value network. To do so, the enterprise need to on-board partners and even the smallest players must be enabled to be on-boarded by bigger ones

To begin the sharing information process, the engaged party (partner, supplier, application developer) need to onboard as partner of the ecosystem owner.

This means that a partner wanting to offer services in the Digital ecosystem needs to be registered and to sign appropriate agreements for the type of services they want to offer. Partners may be Suppliers (of services or content), Developers (of APPs), Resellers (bundling services) or Enterprises.

The intention for partner onboarding in the Digital Ecosystem is to have a lightweight approach similar to an end user signing on to terms and conditions for downloadable application.

So, in a digital ecosystem, the partner will utilize a partner portal which will allow:

- Defining the roles it will play as an engaged party
- Signing to above mentioned agreements
- Creating accounts for billing/charging or settlements related to the selected or offered services.
- Declaring on which account the agreements will apply

In addition, enterprises looking for partnership may wish to share catalog content &catalog management related business processes.

They can act as a Source of catalog information or a Client of Catalog information or both Source and Client.

- A “Source” catalog allows external access to its catalog and business processes and generally restricts or segregates its catalogs and processes for external access.
- A “Client” catalog will discover catalog information and business processes from external Source Catalogs and generally incorporates the discovered information into its own catalog and business processes to form hybrid product offers.

Catalogs can be both a Source to one company and a Client to another, or may be both Source and Client to each other in a mutual bidirectional arrangement

The following three classifications are used to describe these relationships:

- Single Direction Client Catalog - Discover catalog information from a Source Catalog, use portions of the Source company’s fulfillment process.
- Single Direction Source Catalog - Share catalog information with a Client Catalog, allow the Client Catalog to access catalog information and use the Source Catalog’s fulfillment, assurance, charging service management processes.
- Bidirectional Client/Source - Both companies can discover and share information by forming Client and Source relationships with each other. Each company shares and discovers catalog information and each allows the other to utilize fulfillment, assurance, charging service management processes.

The diagram below summarizes the initial high level steps of on-boarding process.

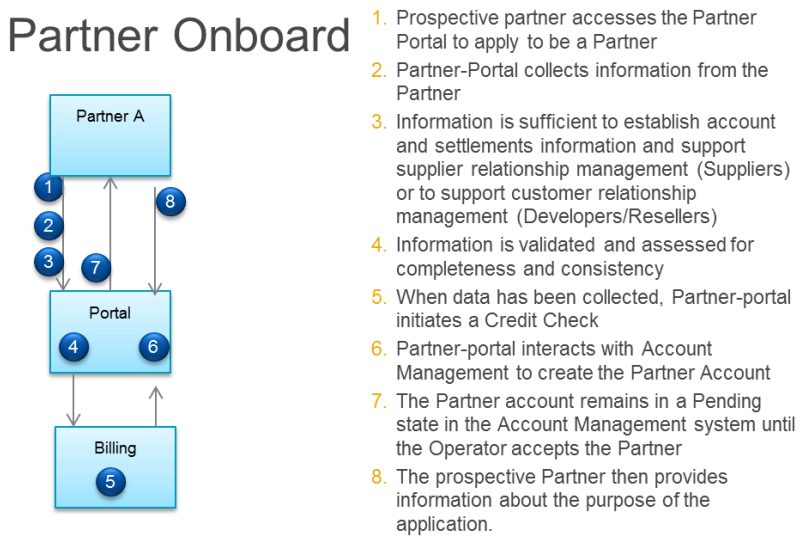

Figure 4: High level view of partner on-boarding process

In conclusion, a typical partner on-boarding use case is detailed below.

Use Case Description:

- **Pre-Condition:** None.
- **Use Case:** Demonstrate automatic on-boarding of partner/supplier/app developer.
- **Post-Condition:** Corresponding Partner is successfully on-boarded in partner management system and billing/charging system.

Use Case Flow:

1. Prospective partner accesses the Partner Portal to apply to be a Partner
2. Partner-Portal collects information from the Partner
3. Information is sufficient to establish account and settlements information and support supplier relationship management (Suppliers) or to support customer relationship management (Developers/Resellers)
4. Information is validated and assessed for completeness and consistency
5. Optionally, when data has been collected, Partner-portal initiates a Credit Check
6. Information collected from partner is utilized to set up partner account
7. The Partner account remains in a Pending state until the Operator accepts the Partner
8. The prospective Partner then provides information about the purpose of the asset they want to contribute: application, web service, API, content.
9. For Engaged Party Role - supplier:
    a. Partner-Care assesses the types of services the Supplier wants to offer and selects an Agreement Template for the appropriate Business Relationship agreements.
    b. The Agreement Template is also used to set access privileges for the Supplier.

10. For Engaged Party Role - Developers:
    a. The type of APP and the expected use of services will drive the choice of Agreement Template.
    b. Developers that are sponsored by the Operator must have a special designation that enables them to select special services that apply only to that relationship.
11. For Engaged Party Role Enterprise :
    a. The Enterprise also has a set of agreements that have to be signed.
    b. In the case of the Enterprise, who brings together the appropriate Ecosystem, Operator Sales may assist in this process.
    c. Enter agreement types from B2B2X partnering guide
12. The prospective Partner electronically signs the appropriate Business Relationship agreements.
13. Partner agreements are retained
14. Partner Agreements are stored electronically and accessible to the Operator Supplier Management via the Partner records
15. Operator Supplier Management may need to be notified of the agreement signing for review.
16. Operator reviews prospective Partner information and Agreements
17. Operator reviews Credit Check Results
18. Operator Approves Partner
19. The partner is now part of the ecosystem
20. Partner-self-care notifies Partner of Approval
21. Partner accesses Partner-self-care to establish credentials
22. Partner ID is generated and sent to Partner
23. Credentials are established for the Partner and retained
24. Partner is enabled for further processing
25. If the Partner is a Supplier, then the credentials enable the Supplier to On-Board a Service though the Self-Care capability Or via catalog management API (e.g. stored in catalog with a non-active PLM state)
26. If the Partner is a Developer, then the credentials enable the Developer to Register an APP through Self-Care or to purchase product offerings through the portal storefront
27. If the Partner is a Reseller, then the credentials enable access to create a bundled offering or purchase an API with a resulting mash up.
28. If the Partner is an Enterprise, then the credentials enable the enterprise to submit partner names to ecosystem.
29. Partner uses credentials to log in and review catalog information.

Post Conditions

- Partner account has been created
- Partner relationship records have been created
- Partner information to support electronic settlements has been established
- The Partner is enabled as follows:
    - Suppliers are enabled to On-Board Services via Partner-self-care
    - Developers are enabled to purchase product offerings, develop and register an APP

- Enterprises are enable to submit partners, utilize the B2B Storefront and select Operator Services
- Resellers are enabled to access the B2B Storefront for Service Selection.

2.3. A typical business scenario example

Leveraging the above described customer journey, the business scenario presented below details:

- The actors involved in it
- The interactions (touch points) between them during digital service lifecycle
- The API used to support interactions

In this use case, Partner A has the commercial relationship with final customer and partner B provides a product or service that partner A resells to final customer.

Actors involved in this business scenario are:

- Customer
- Partner A with role of service provider
- Partner B with role of service provider supplying service B to partner A

Figure 3 provides a detailed view of interactions and API used

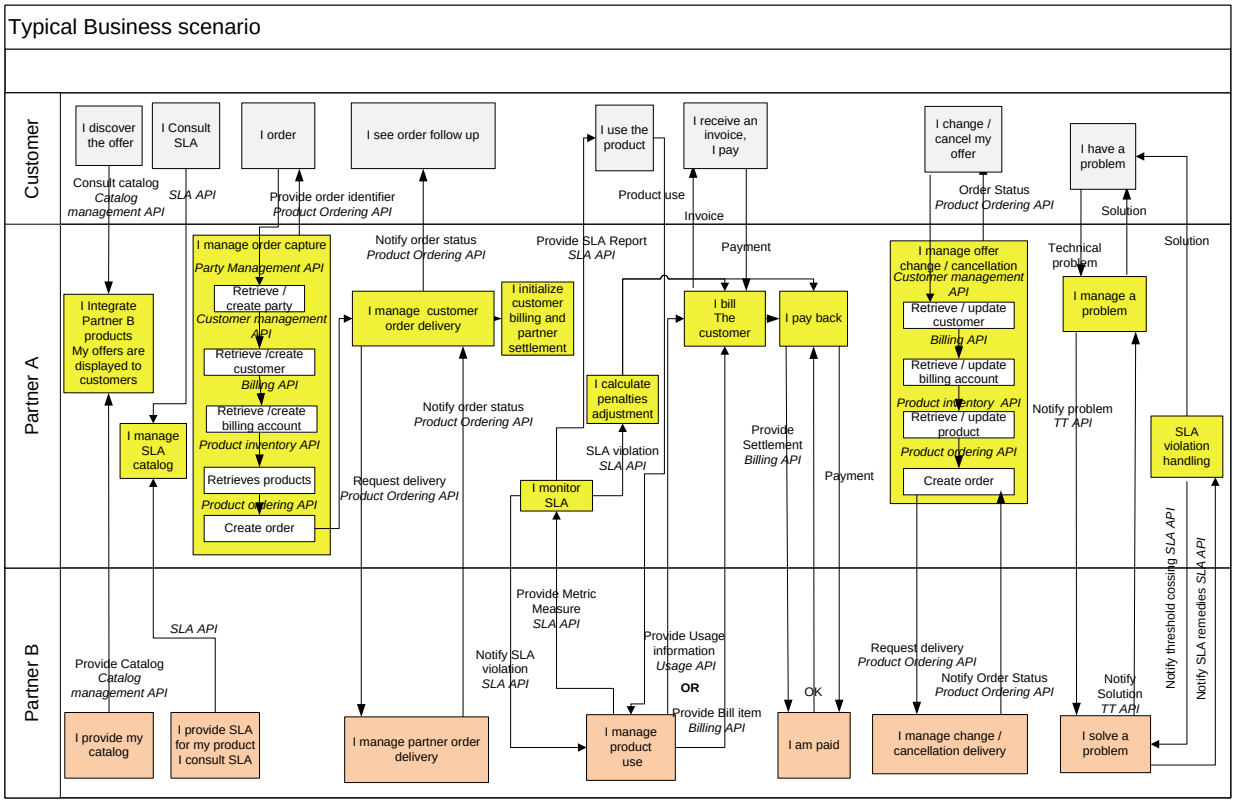

Figure 5 - detailed interactions and related API

Note: for the sake of clarity, offer cancellation has been removed and merged with offer change.

The table below summarizes:

- The actors involved
- The touch points between actors
- The API used
- The operation performed
- The ODE business entity managed

Some touch points don't appear in the table:

- Payment from customer as payment API is not yet specified
- Payment to partner as payment API is not yet specified
- Product use as it involves functional (service usage) APIs
- Customer Invoicing as Invoicing API is not yet specified
- Settlement note validation as no API is specified yet for such interaction
- Technical problem submission by customer as it is not realized using API

Actor 1	**Actor 2**	**Touch point**	**API**	**Operation**	**ODE business entity managed**
Customer	Partner A	Consult Catalog Customer consults catalog from partner A	Catalog management	Catalog retrieval	Product offering
Partner B	Partner A	Provide catalog Partner B provides catalog to partner A Partner A integrates partner B products	Catalog management	Catalog export	Catalog
Customer	Partner A	Consult SLA Customer consults SLA of Partner A products	SLA management	SLA retrieval	SLA
Partner B	Partner A	Provide SLA Partner A provides SLA of its Products	SLA management	SLA update	SLA
Partner B	Partner A	Consult SLA Partner A consults SLA of Partner A products	SLA management	SLA retrieval	SLA
Customer	Partner A	Place Order Customer places order to partner A Partner A retrieves / create party Note: this step allow creation of an individual or organization with its identification information and contacts	Party management	Retrieve party and create party if party doesn't exist	Party
Partner A	Partner A	Retrieve / create customer Partner A retrieves customer and creates it if necessary Note: this step allows creation of a customer role to party already created	Customer management	Retrieve Customer and customer account Create customer and customer account if customer doesn't exist	Customer Customer account
Partner A	Partner A	Create / retrieve billing account	Billing	Retrieve and Create	Billing account

Actor 1	Actor 2	Touch point	API	Operation	ODE business entity managed
		Partner A retrieves billing account and creates it if necessary		billing account if billing account doesn't exist	
Partner A	Partner A	Retrieve product Partner A retrieves product from product inventory	Product inventory	Retrieve existing products	Product
Partner A	Partner A	Create order	Product ordering	Create customer Order	Product order
Partner A	Partner B	Request delivery Partner A creates order to requests order delivery to partner B	Product Ordering	Create partner Order	Product order
Partner B	Partner A	Notify order status Partner B notify order status to partner A	Product Ordering	Notify Order Status	Product order
Partner A	Customer	Notify order Status Partner A notify order status to customer	Product Ordering	Notify Order status	Product order
Partner B	Partner A	Provide Usage information Partner B provides usage information to partner A	Usage	Provide usage	Usage
Partner B	Partner A	Provide metrics measurement Partner B provides Metric measurement information to partner A	SLA management	SLA notification	SLA
Partner A	Customer	Provide SLA report Partner B provides SLA report to customer	SLA management	SLA notification	SLA
Partner A	Partner B	Notify SLA violation	SLA management	SLA violation notification	SLA violation
Partner A	Partner A	Retrieve SLA Violation Partner A retrieves SLA violation to Calculate penalties and adjustments	SLA management	SLA violation retrieval	SLA violation
Partner B	Partner A	Provide Bill item Partner B provides bill item to partner A	Billing	Send bill item	Bill item
Customer	Partner A	Send Payment Customer pays Partner A	None	No payment API	None
Partner A	Partner B	Provide Settlement Partner A settles with partner B	Billing	Notify settlement note	Settlement note advice
Customer	Partner A	Change Offer Customer requests offer change to partner A Partner A retrieves customer information (note: it is not necessary to retrieve party as customer already exists)	Customer management	Retrieve customer / customer account	Customer Customer account
Partner A	Partner A	retrieve billing account Partner A retrieves billing account	Billing	Retrieve billing account	Billing account

Actor 1	Actor 2	Touch point	API	Operation	ODE business entity managed
Partner A	Partner A	Retrieve existing products Partner A retrieves customer existing products	Product inventory	Retrieve /update existing products	product
Partner A	Partner A	Create order Partner A creates customer order	Product Ordering	Create customer order	Product order
Partner A	Partner B	Request delivery Partner A creates partner order to request order delivery to partner B	Product Ordering	Create partner order	Product order
Partner B	Partner A	Notify order status Partner B notifies order status to partner A	Product ordering	Notify Order status change	Product order
Partner A	Customer	Notify order Status Partner A notifies order status to customer	Product Ordering	Notify order status change	Product order
Partner A	Partner B	Notify problem Partner A notifies problem to partner B	Trouble ticketing	Create trouble ticket (internal) Notify trouble ticket	Trouble ticket
Partner B	Partner A	Notify solution Partner B notifies solution to partner A	Trouble ticketing	Update trouble ticket (internal) Notify trouble ticket update	Trouble ticket
Partner A	Partner B	Notify SLA threshold crossing Partner A Notifies SLA threshold crossing to partner B	SLA management	SLA Violation notification	SLA violation
Partner B	Partner A	Notify SLA remedies Partner B Fixes the problem and notifies SLA remedies to partner A	SLA management	SLA Violation update	SLA violation

3. Business Scenario examples

This section:

- Details instantiated business scenarios industry specific and coming from real-life.
- Presents one E2E use case for a given industry (ordering – delivery – billing – TT etc. for automotive) and not several separate use cases with different scopes as presented during specification jam (i.e. automotive for Ordering /Billing, Ehealth for TT, etc.)
- Presents an additional use case focused on Smart Cities domain

Description includes:

- Literary description of the business scenario
- Standard diagrams presenting sequence of operations and interactions between players.

3.1. Connected car example

3.1.1. Scenario overview:

At the beginning, connectivity in cars was limited to safety related services such as e-call (automatic emergency call), b-call (roadside assistance) and stolen vehicle tracking. Now car manufacturers want to address infotainment needs. Connectivity offers a seamless access to services and contents.

Car manufacturers may be afraid of driver infotainment data consumption volume and may not want to pay for related charges. A solution consists to split the service charging between the car manufacturer and the driver (who becomes a third-party payer).

In this situation, the car manufacturer remains the owner of the ecall SIM card and the infotainment SIM card (note that depending on models, both services can be supported by a unique SIM card). The car manufacturer only pays for charges for the ecall service data consumption and declares the driver as a third-party payer for the charges of the infotainment data consumption. The driver may already be a CSP (Communication Service Provider) customer or not. The car manufacturer, being the SIM card(s) owner, remains the only liable party in case of default.

The car manufacturer either handles the third party payer themselves or gives a mandate to CSP to perform this management. In the latter case, the driver will use a CSP portal to update his information such as bank account information, billing address…

The car manufacturer can outsource connected car customer support to a call center which:

- takes calls from drivers when a problem occurs,
- analyses them,
- distributes them to car manufacturer or CSP according to their nature,
- Provides an answer to customer when problem is solved.

The call center fee depends on the number of handled trouble tickets and several processing quality indicators of these tickets. In addition, the car manufacturer may invoice CSP depending on number of trouble tickets related to connectivity issues

The call center can also monitor / audit SLA and handle SLA violation by notifying SLA violation events to the car manufacturer and the CSP.

Car manufacturer can also partner with infotainment service provider (or aggregator) and CSP to create new infotainment converged offer. These offers are managed in a central catalog accessible by partner and can be ordered through different distribution channels (car manufacturer, CSP or infotainment content provider).

3.1.2. Actors involved

- Car manufacturer
- Communication Service Provider (CSP)
- Final customer (driver)
- Content provider (for infotainment)
- Call center (for customer support)

The diagram below presents the high level interactions between actors

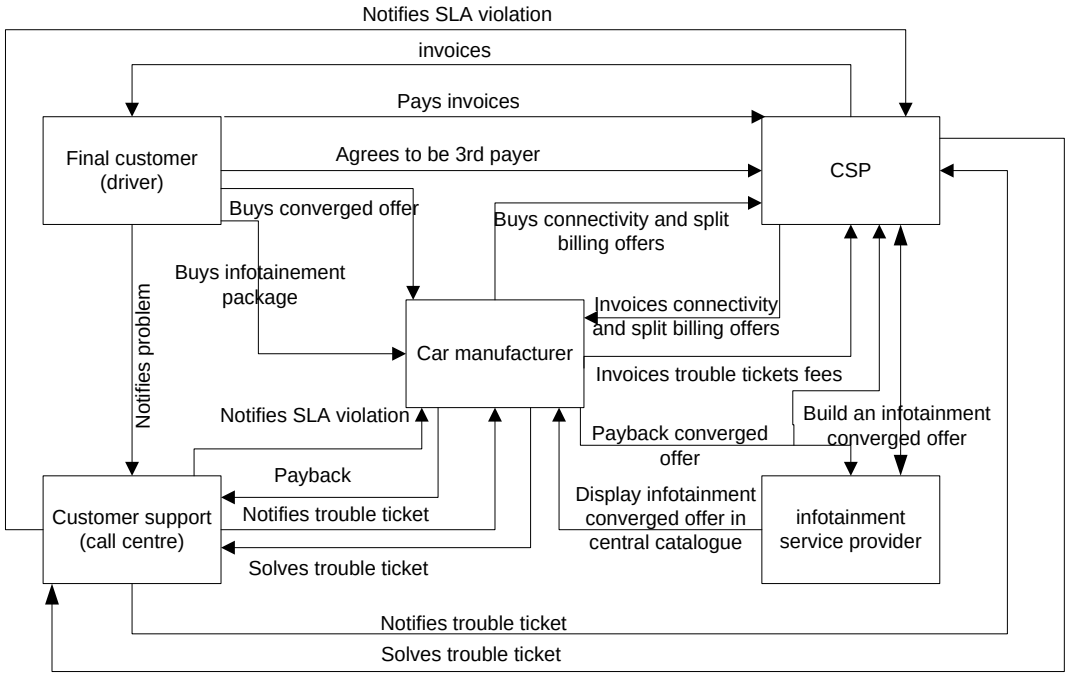

Figure 6 – Connected car high level interactions

3.1.3. Scenario detailed description

Part 1: Product Setup and billing

- Car manufacturer buys from the CSP
  - a connectivity offer
  - a split billing offer
- CSP provides to car manufacturer all needed technical information to integrate CSP management API (web portal credentials, name of the API bundle corresponding to the subscribed offer),
  - Note: an “API bundle” is a set of API (such as Ordering API, billing API etc.) corresponding to the offer subscribed by the partner that it will use to interact with CSP
- The car manufacturer integrates the API with their own IT system and proceeds to their deployment
- The CSP registers the offer subscription in the billing system

- The car manufacturer retrieves the infotainment data connectivity offer from CSP catalog by using *catalog management API* (probably with some filtering criteria to access only the part of the catalog corresponding to his needs) and builds its own service catalog based on connectivity offers provided.
- A the end of billing period, CSP rates offers fees and produces an invoice including connectivity and split billing fees
    - Car manufacturer receives an invoice
    - By using *Billing API*, Car manufacturer can also receive bill items corresponding to its invoice for invoice control or consumption follow up purposes

The diagram below summarizes the sequence of actions

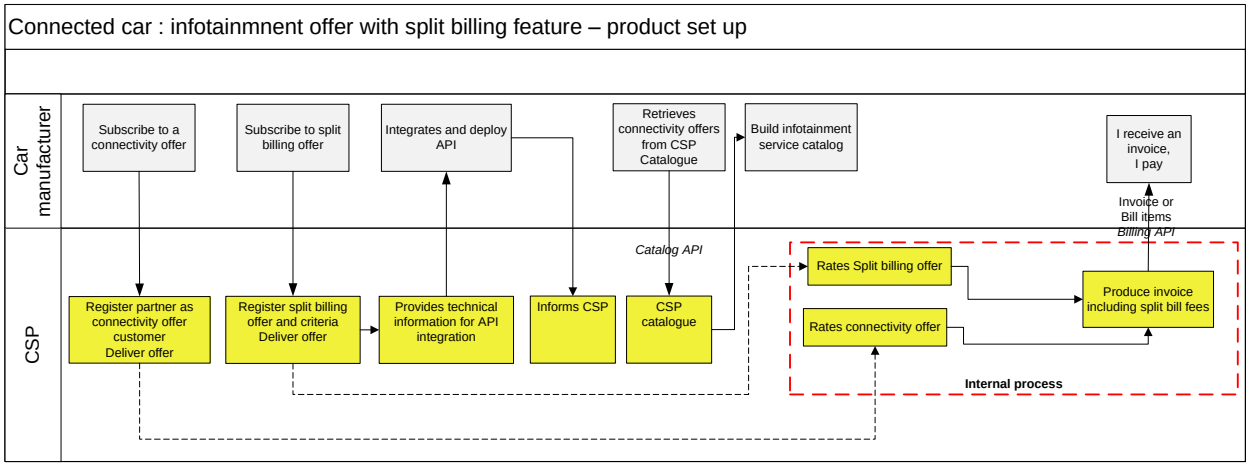

Figure 7 – Connected car product setup and billing

Part 2: Service Use

- When final customer (driver) buys a car, car manufacturer offers them an infotainment service associated with a connectivity service billed by CSP
- Driver selects an infotainment service and agrees to be 3rd party payer for connectivity service as connectivity is not included in infotainment service cost.
- Car manufacturer
    - Registers infotainment service order in its own system
    - Checks if driver is already a CSP customer by using *Customer management API* and, if so, retrieves customer billing information by using *Billing API*
    - Orders the infotainment data connectivity CSP offer
        - *Party API* allows car manufacturer to provide CSP with information allowing it to create 3rd payer
        - *Product ordering API* allows car manufacturer to place order for infotainment connectivity package
- CSP creates:
    - 3rd party payer,
    - Order for infotainment connectivity offer (for advanced offers, it will be possible to detail split billing criteria)
- CSP asks its mediation system to send usage related to the subscription
- Driver uses infotainment service, which generates data usages to bill on infotainment package

- *Usage API* allows car manufacturer to retrieve usage for consumption follow up purposes

- CSP:
  - Splits infotainment usages from telematics usages
  - Rates infotainment usages for 3rd payer
  - Rates telematics usages for car manufacturer
- At the end of billing period, CSP:
  - Rates recurring fees for 3rd payer and car manufacturer
  - Produces invoices for car manufacturer and 3rd payer
- By *using Billing API*, car manufacturer can also receive bill items corresponding to its invoice

The diagram below summarizes the sequence of actions

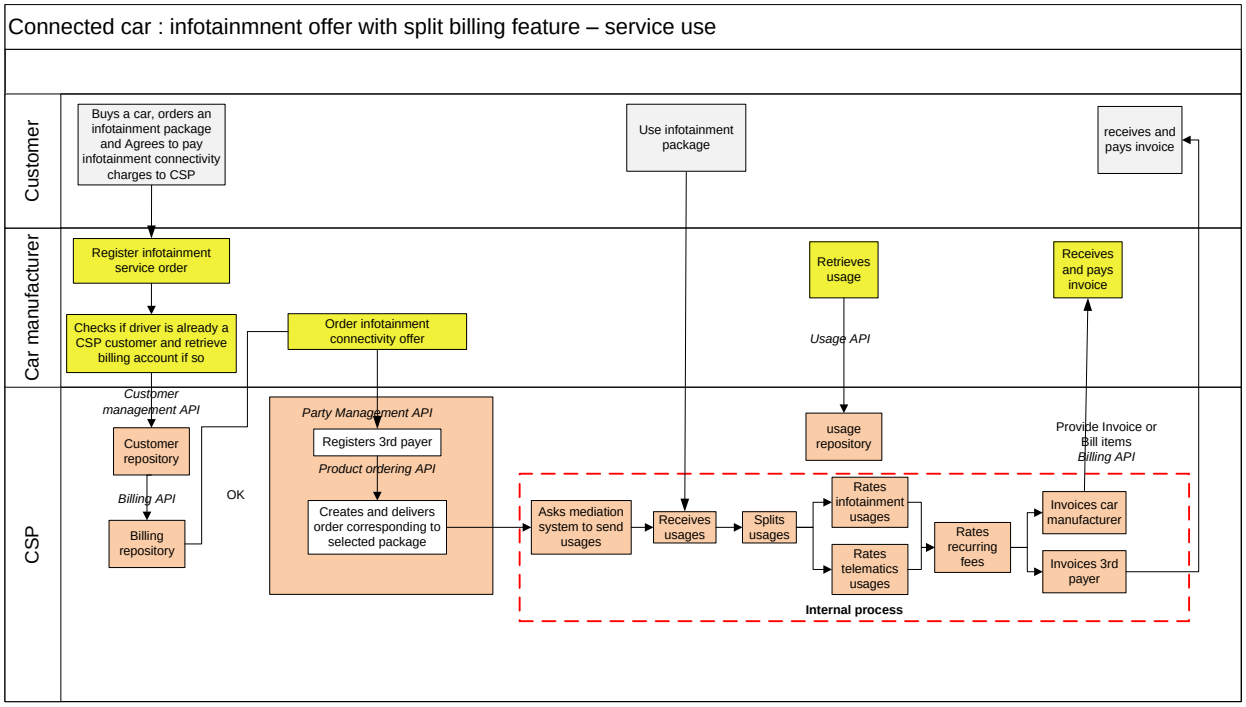

Figure 8 – Connected car service use

Part 3: trouble ticket management

Note: this part is based on the" Trouble ticketing management in ehealth domain" use case presented within the "ABC of multi-party digital services" catalyst adapted to automotive context.

- The call center manages all technical issues coordinating their resolution either by CSP or car manufacturer
- When receiving a support request from a driver, call center uses *Trouble Ticket API* to:
  - create a trouble ticket

    - send trouble ticket to CSP or car manufacturer according to nature of the trouble

- CSP or car manufacturer analyses the trouble and can redirect trouble ticket in case of misdirection by using *Trouble ticketing API*
- *Trouble ticketing API* allows also CSP / car manufacturer to update trouble ticket status /information depending on problem resolution progress and to notify updates to call center
- When problem is solved, CSP / car manufacturer uses *Trouble ticketing API* to send back Trouble Ticket to call center
- Call center informs customer and closes the trouble ticket
- CSP information concerning the trouble tickets is sent to car manufacturer which will use it to calculate the call center settlement note
- Car manufacturer sends settlement note advice to call center by using *Billing API*
    - Call center accepts or disputes the settlement notes
    - If call center agrees, CSP pays the settlement notes amount
- Car manufacturer may invoice CSP according to number of trouble tickets related to connectivity issues and send bill items corresponding to invoice by using Billing API

Call center can also play the role of SLA monitor / auditor:

- Call center can subscribe to SLA violation events notification by using *SLA management API*
- By using *SLA management API*, Call center can also :
    - Provide customer with SLA reports about SLA violation
    - Notify SLA violation creation to car manufacturer and CSP
- CSP and car manufacturer can retrieve SLA information by using *SLA management API*
- Call center can send alarms to teams in charge of problem resolution by using *Trouble Ticketing API*
- CSP and car manufacturer can notify SLA violation to the team in charge of problem resolution with related metric identifier by using *SLA management API*
- Teams in charge of problem resolution can correlate alarms with metric identifier
- End of SLA violation resolution process is, then, managed within trouble ticketing process

The diagram below summarizes the sequence of actions:

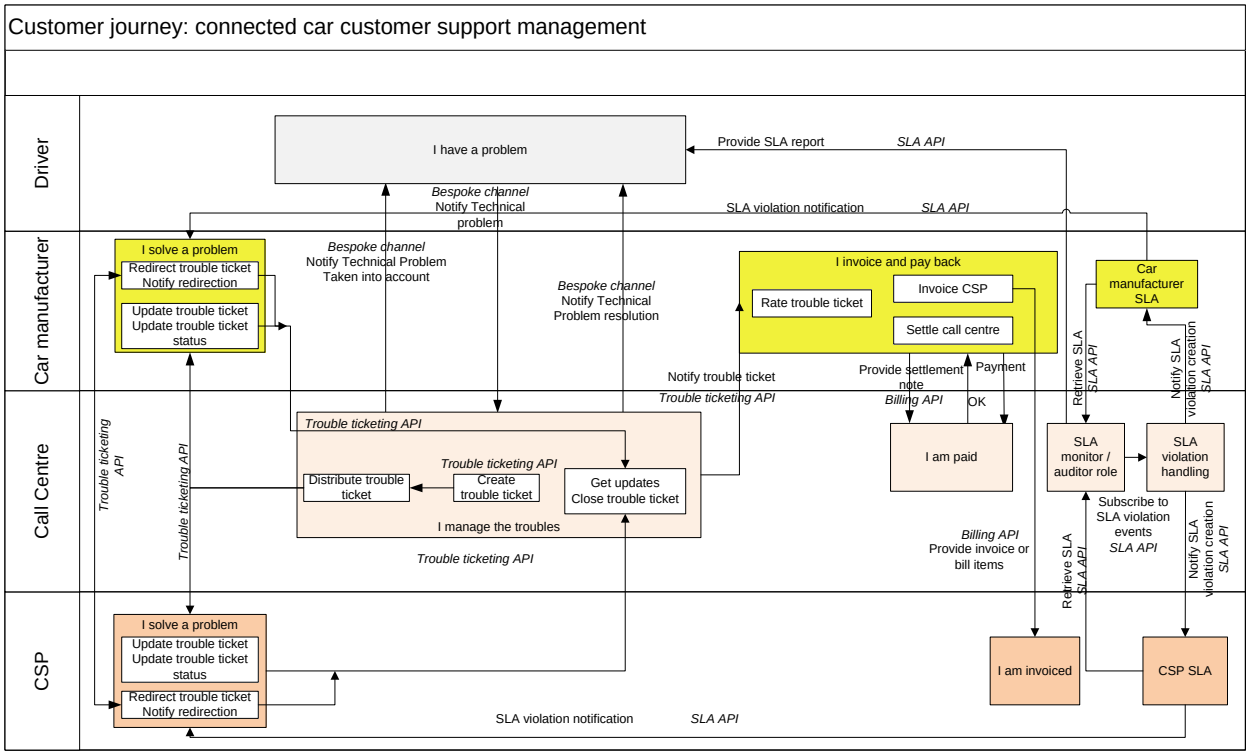

Figure 9 – Connected car trouble ticket and SLA management

Part 4: Creation and ordering of a new converged infotainment service

Note: this part is based on the" Digital service provisioning in a multi partner context" use case presented within the "ABC of multi-party digital services" catalyst adapted to automotive context.

- Creation of a new converged infotainment service
    - The CSP Catalog Manager defines CSP products which are available in central catalog managed by car manufacturer, for inclusion in 'converged' offers:
        - CSP Catalog Manager creates a product offering (for example 5 Gb data download for X euros)
        - CSP Catalog Manager marks the offer eligible for partners
        - By using *catalog management API*, CSP Catalog Manager sends the product to central catalog so that partners can choose the products to include in their 'converged' offers
    - The infotainment content provider (or content aggregator) defines a new converged offer:
        - Infotainment content provider creates a new offer (for example access to musical contents)
        - By using *catalog management API*, infotainment content provider retrieves CSP offers from central catalog

        - Infotainment content provider creates a new converged offer (for example 5 Gb music download for Y euros)
        - By using *catalog management API*, infotainment content provider can register its new converged offer in central catalog
    - By *using catalog management API*, CSP and car manufacturer can retrieve the infotainment converged offer and verify its correctness
    - By using *catalog management API*, the infotainment converged offer is sent to CSP, infotainment service provider and car manufacturer product catalogs so that customer can order it by using any of the three channels

- Ordering / delivery of the infotainment service
    - The customer (driver) places an order for a digital service via car manufacturer self-service portal and car manufacturer registers the order
        - *Customer management API* is used to retrieve customer and customer account
        - *Billing API* is used to retrieve billing account and create it if it doesn't exist
        - *Product inventory API* is used to retrieve existing products (if any) and check compatibility
        - *Product ordering API* is used to create infotainment converged offer order
    - Sometimes but generally later, customer requests service activation to infotainment service provider
        - By using *Product inventory API*, infotainment service provider checks if concerned service is owned (has been ordered) by the customer
        - If so, infotainment service provider uses *Product ordering API* to request specific connectivity service activation to CSP
        - By using *Product ordering API* , CSP notifies connectivity service order realization to infotainment service provider
    - infotainment service provider activates the service and use *product inventory API* to notify car manufacturer
    - At the end of the billing cycle, car manufacturer:
        - produces an invoice for customer
        - calculates settlements and uses *Billing API* to payback infotainment service provider and CSP

The diagrams below summarize the sequence of actions:

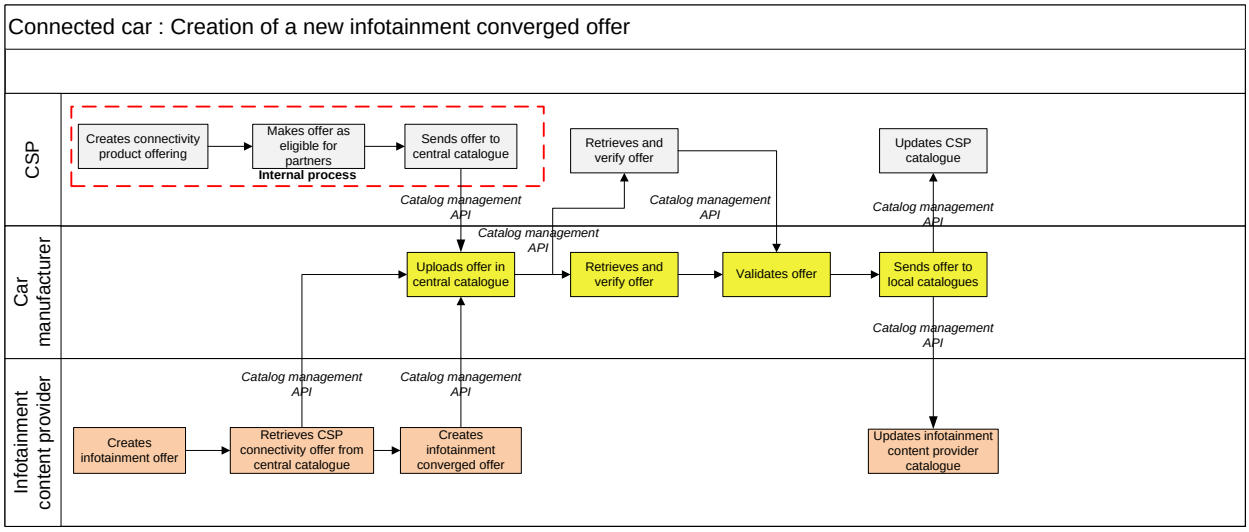

Figure 10 – Connected car creation of a new offer

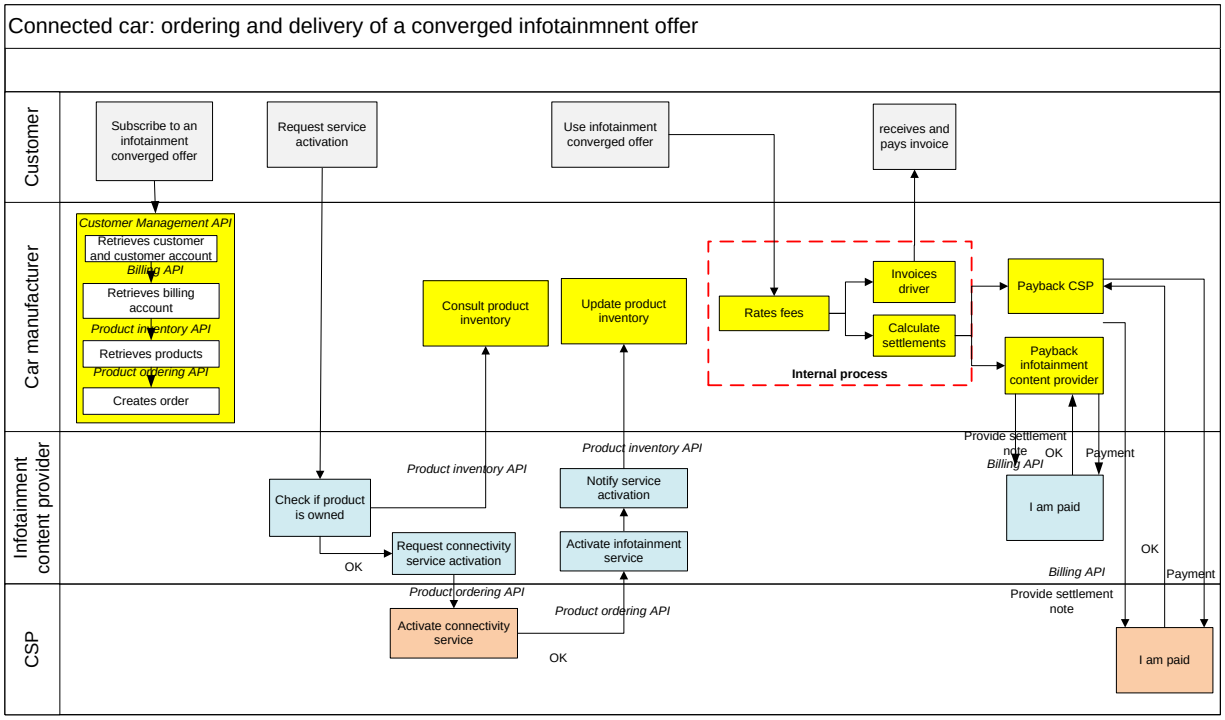

Figure 11 – Connected car ordering and delivery of a new offer

3.2. Smart Cities example

3.2.1. Scenario overview

The goal of this scenario is to provide Internet on board via an embedded connectivity solution associated to a ground network and a service platform

To do what?

- B2C customers: to offer to public transport users a high quality internet connection with dedicated consumers services (travellers information and leisure)
- B2B customers: to differentiate from the competition by offering services requested by travellers and pooling investments related to legal constraints (e.g. information for hearing impaired passengers)

Process description

- Step 1: internet on board sponsoring
    - CSP sells an internet on board connectivity offer to a public transport company
    - CSP sets up a voucher system to sponsor internet use in public transport and encourage to use them
        - CSP generates a voucher for B2C customers who use it as discount when purchasing internet on board service.
            - Voucher generation can be linked to special connectivity offer
            - Customer can also burn loyalty points to “buy” vouchers
        - Sponsoring can be totally financed by CSP or in partnership with the public transport company
            - If it is totally financed by CSP, there is no impact on invoice issued to the public transport company
            - If it is financed in partnership with public transport company: the invoice issued by the CSP to the public transport company will be reduced by the sponsored amount it has already paid
            - In any case, public transport company checks voucher validity and updates its status to prevent it from being reused
- Step 2: 'profiling‘ option management:
    - CSP proposes to final customer a free offer based on customer agreement allowing it to analyze customer data and provide “profiling” information to partners
        - Final customer indicates which data can be used and for what kind of partner
        - Depending on data and authorized partners types, CSP generates a discount voucher when use of customer data via a partner query reaches a given threshold
    - CSP sells profiling offer to public transport company and is paid each time they make a query related to data included in the offer

- Query realization generates “value points” to final customers whose data have been queried. Then, the public transport company knows that CSP will generate discount vouchers. Final customers will receive these vouchers and will be able to use them to pay transportation services.
- Public transport company knows that the data it queries are related to customers that have given their agreement and are interested in its services => its queries are more focused
- Final customers will use vouchers for public transport company services. Public Transport Company can analyze the used voucher rate compared with requests it has made and, optionally, can adapt voucher nature. As mentioned above, vouchers are checked and their status updated

3.2.2. Actors involved

- Communication Service Provider (CSP)
- Public transport company
- Final customer (traveler)

The diagram below presents the high level interactions between actors

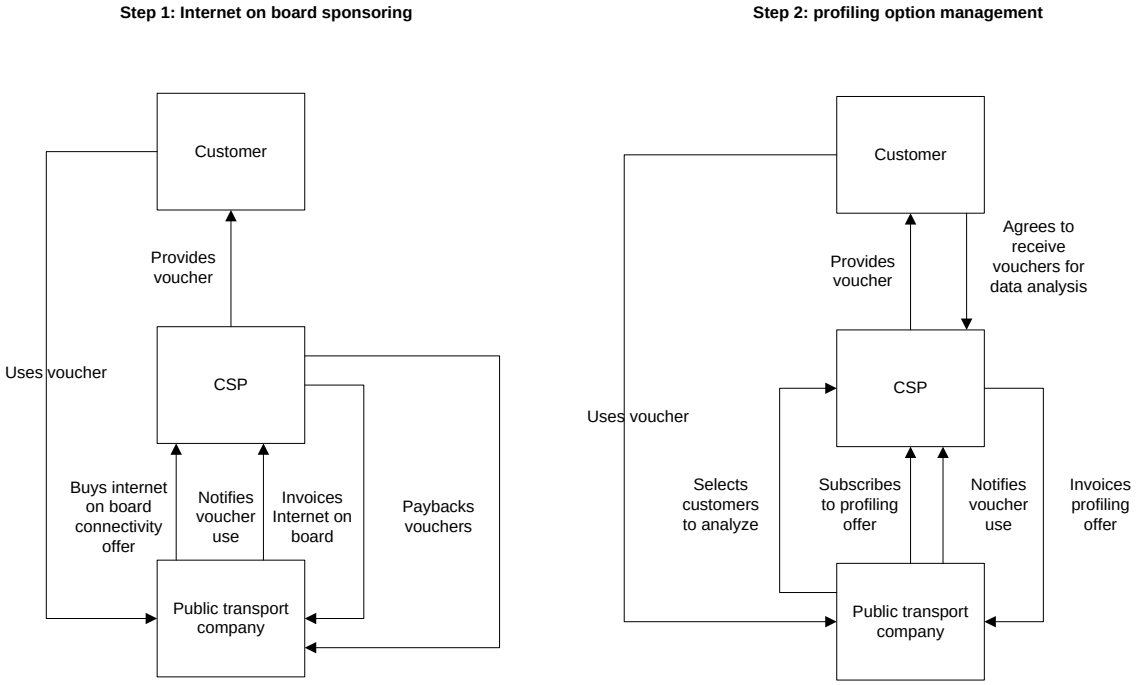

Figure 12 – Internet on board high level interactions

3.2.3. Scenario detailed description

Internet on board sponsoring managed in partnership between public transport company and CSP

- Public transport company buys a connectivity offer for internet on board from the CSP
- CSP delivers the internet on board offer
- CSP provides to public transport company all needed technical information to integrate CSP API (web portal credentials, name of the API bundle corresponding to the subscribed offer

- The public transport company integrates the API with its own IT system and proceeds to their deployment
- CSP generates a voucher and provides it (push mode) to final customer
    - Voucher is considered as a “free” offer
    - CSP registers voucher in its product inventory with a validity period
- Final customer buys a transport ticket including internet on board service to public transport company using voucher
- Public transport company sells ticket and registers voucher
    - By using *Product Inventory API*, public transport company can check voucher validity and update CSP product inventory to indicate that voucher has been used
- At the end of billing period, CSP:
    - Rates internet on board offer
    - Rates voucher
    - Can
        - Invoice public transport company subtracting voucher amount from invoice total amount
            - By using *Billing API*, public transport company can receive bill items corresponding to its invoice
        - **OR** Invoice public transport company and calculates settlements to payback public transport company
            - By using *Billing API* , CSP can send settlements note advice to public transport company

The diagram below summarizes the sequence of actions:

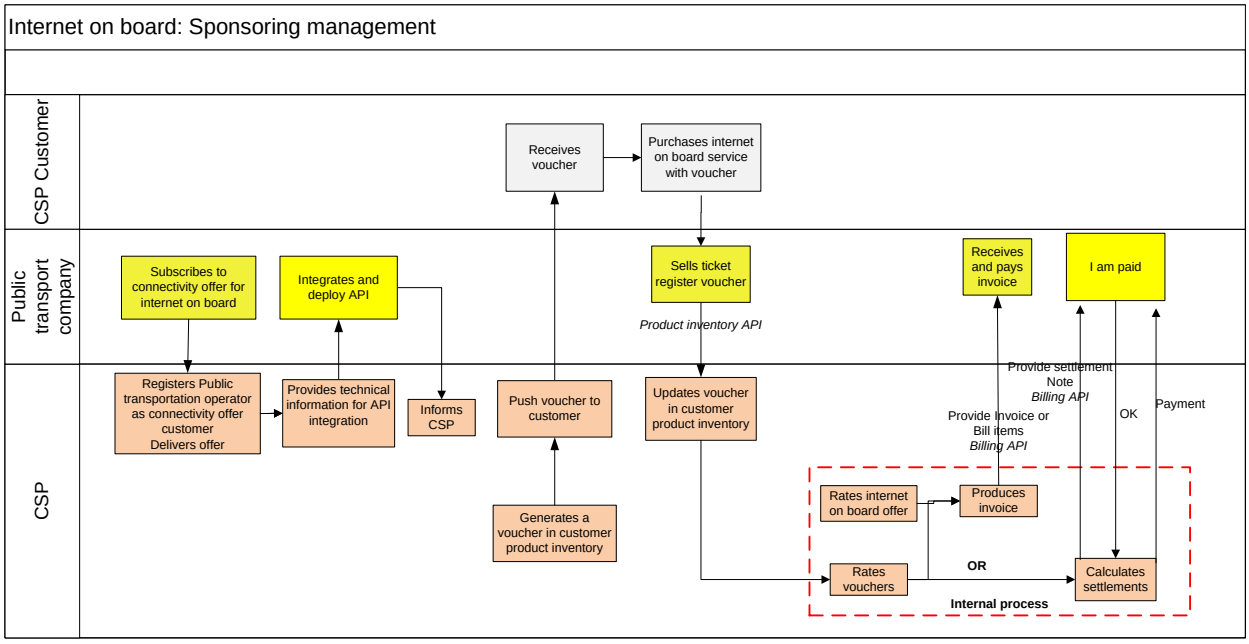

Figure 13 – Internet on board sponsoring management

Profiling option management

Assumption: There is no payback process from CSP to public transport company related to voucher use because cost of voucher is supposed to be included in profiling offer cost

- CSP proposes to final customer a free offer allowing CSP to provide customer data to public transport company for analysis purposes in exchange for vouchers
    - Final customer agrees and indicates which data can be used and for what kind of partner
    - CSP updates its product inventory
- CSP proposes profiling offer to public transport companies
- Public transport company buys the profiling offer from the CSP
- CSP delivers the profiling offer
- Public transport company uses offer and selects customer (explicit list or corresponding to given criteria) to be queried
    - *Product inventory API* is used to update public transport company product inventory
- Public transport company performs queries
    - Beyond a given threshold, CSP generates a discount voucher in its product inventory with a validity period and provides it to final customer
- Final customer buys a service to public transport company using voucher
- Public transport company sells service and registers voucher
    - By using *Product Inventory API*, public transport company can check voucher validity and update CSP product inventory
- At the end of billing period, CSP:
    - Rates profiling offer
    - Invoices public transport company
        - By using Billing API, public transport company can receive bill items corresponding to its invoice

The diagram below summarizes the sequence of actions:

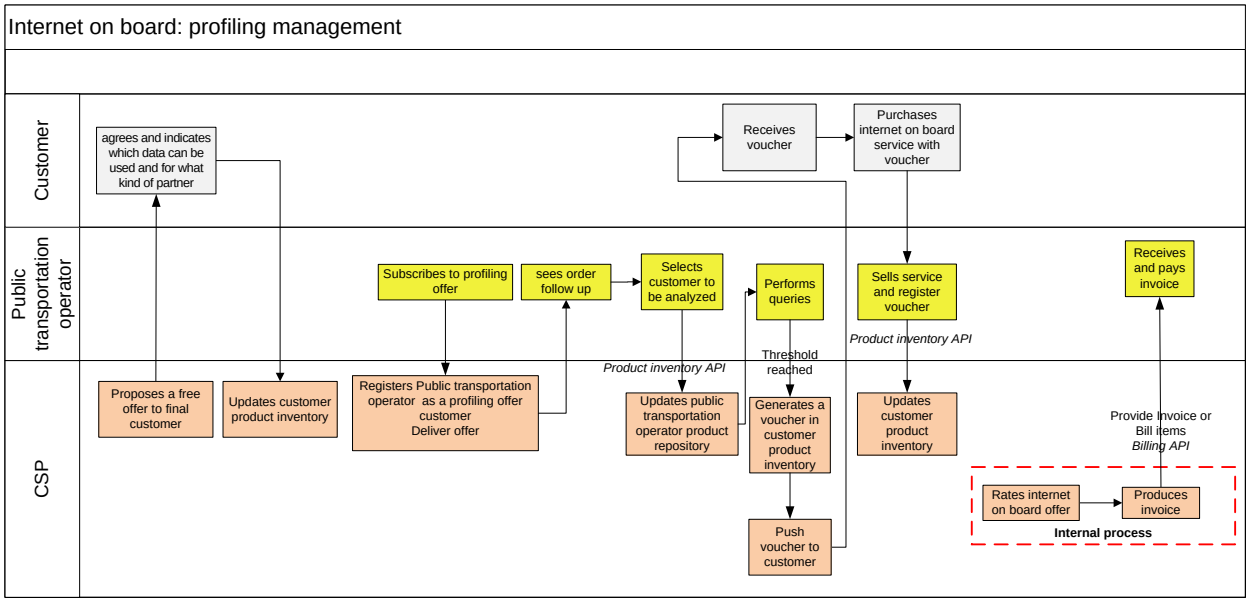

Figure 14 – Internet on board profiling option management

4. Generic Pattern for each API

This document contains already three business scenarios that show how the API's can be used today to deliver a value proposition to the customer. We can extend this document with hundred other examples in other industries to each address another unique angle in the digital economy. These examples above helps the business & developer community tremendously to compare what others have done, but still means that each of these steps needs to be drafted from scratch on top of the provided API's.

So often, it is helpful to have the commonalities between these scenarios pre-distilled, which serves as departure point for a whole new business scenario, but still leaves ability to adapt the basic design to support a whole new innovative differentiator. So - in addition to the API's and documentation we also provide Quick-Start Building Blocks – which distill the essence of 100's such scenarios by only keeping the most obvious common patterns. These simple patterns can then be extended to In order to construct a generic Pattern; we'll do a top-down, bottom-up analysis (MIM).

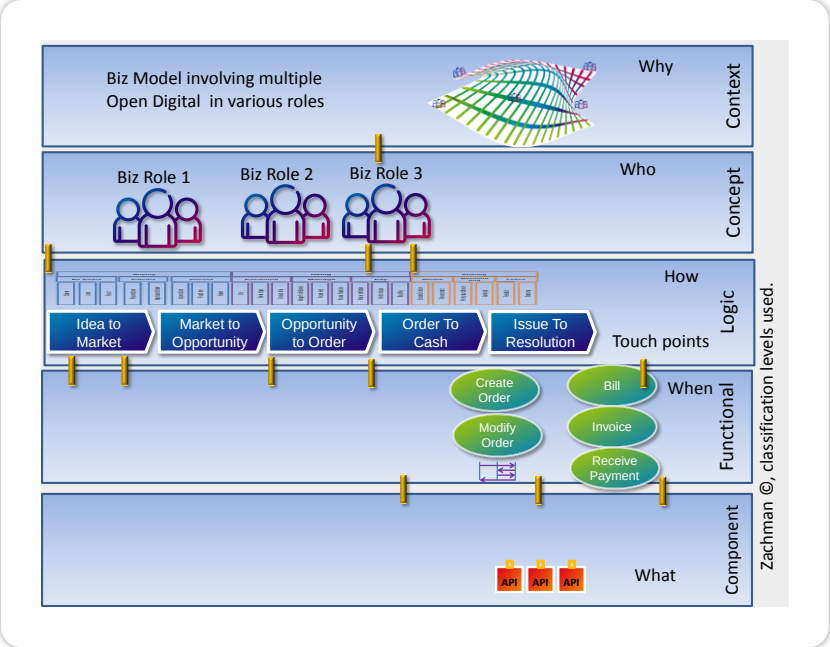

Figure 15 Separation of Interests – Building for reuse

The goal of a generic pattern is to provide quick ramp up building block to all parties involved in the B2B2X relationship. It can be used by the business community to draft new business opportunities in the open Digital space.

By using these Quick-Start Building Blocks, it is easier to discern which part of the own business model, require marginal tweaks before it can be launched. For example we'll list some common business models (subscription Business Model), link this for example to the standard e2e processes (a) Use to Bill and (b) Bill to Payment), show which Use cases apply where on the e2e Processes and use the sequences that belongs to that use case.

In addition to the materials provided in this document, reader will find additional information about:

- "How to" build business scenario involving multiple partners in more information in B2B2X Partnering Accelerator Business Pack and Online B2B2X Partnering Step by Step Guide
- End to End process building blocks patterns (Idea to Launch, usage to pay, problem to solution, etc.) in documents such as GB GB921 appendix E (End-to-End Business Flows) and F(Process Flow Examples.

4.1. Using the Quick Start Building Blocs to quickly ramp up

This section describes how to use Quick-Start Building Blocks to construct everyday business scenarios by applying them to realize value visible to Business Stakeholders and yet allow for different technical subject matter experts working relatively independent towards delivering such business scenario.

It will serve two purposes:

- For the DSP (operator or other player) preparing to offer such TM Forum Open Digital API service in general (nationally / internationally across subsidiaries)
- for the community / party that wants to understand how API are used in conjunction with others, their variations (use cases) and applicability (in end to end processes)

To proof their usability, we can use the example of a standard process such as T2R (trouble to resolve) and show by reference how a generic process may be plugged-in (and modified) to fit another business scenario.

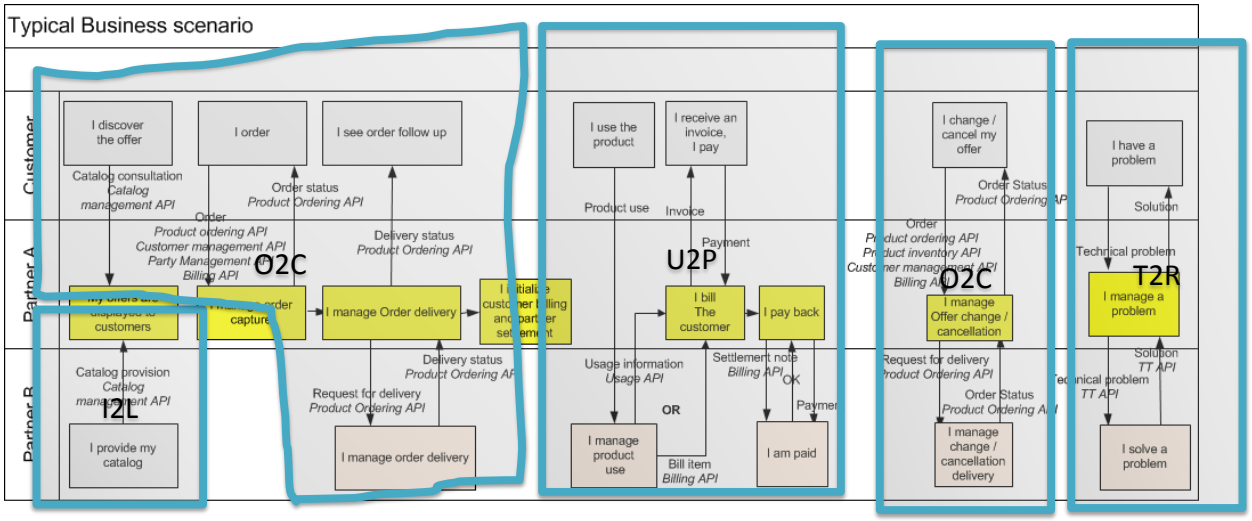

Figure 16 Separation of interests into major e2e processes - Applying e2e processes blocks

We can also quickly evaluate the coverage of an existing business scenario. See for example **Error! Reference ource not found.**, below. It is clear that there might be challenges in rolling out the scenario if A) Idea to Launch, i.e. setting up partnerships upfront, is neglected before a converged offer is built, and B) tracking customer usage and profile information (beyond billing relevant information) might not be sufficient for measuring campaign effectiveness.

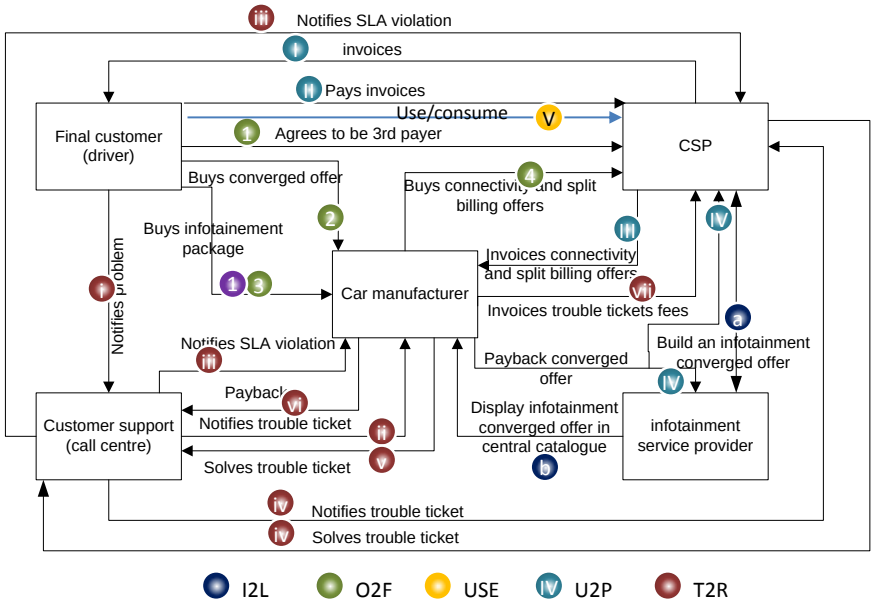

Figure 17 : Applying e2e services in an e2e process onto a Business Scenario

4.2. Some exemplary use case building blocks and sequences

Each of the above mentioned processes contains a number of applicable use cases with various iterations during the process so that multiple variants of a use case (sunny day and rainy day scenarios) are covered. This business guide will concentrate on providing the absolute necessity. The business community might extend this with further additions and variations /exceptions needed for their own business scenario.

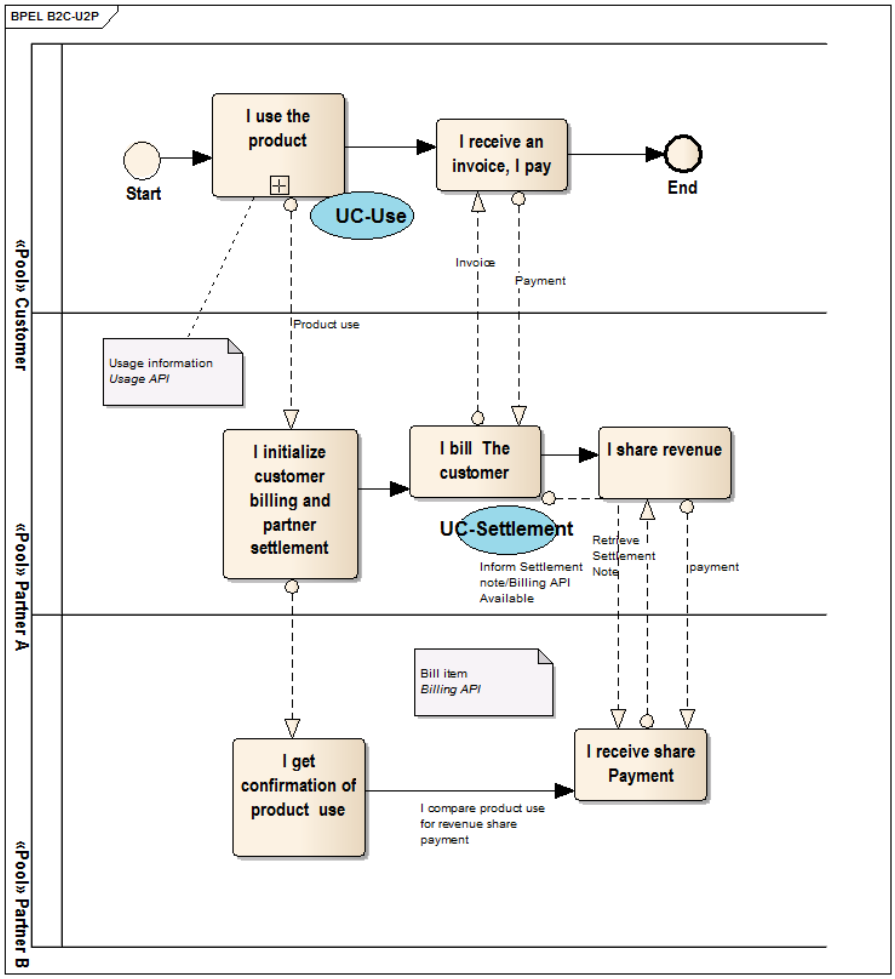

Figure 18 : Generic Quick Start building Block for Usage to Pay process

The above diagram provides a generic Quick Start Building Block for Usage to Pay business process.

Based on this QSBB, the different diagrams below provides generic patterns for the above described interactions associated with REST API calls and resulting actions from both parties.

Sequences which are “not modifiable” (for the sake of sequence consistency) are marked in red. This means that the relative order of procedure between the OD API calls should be maintained.

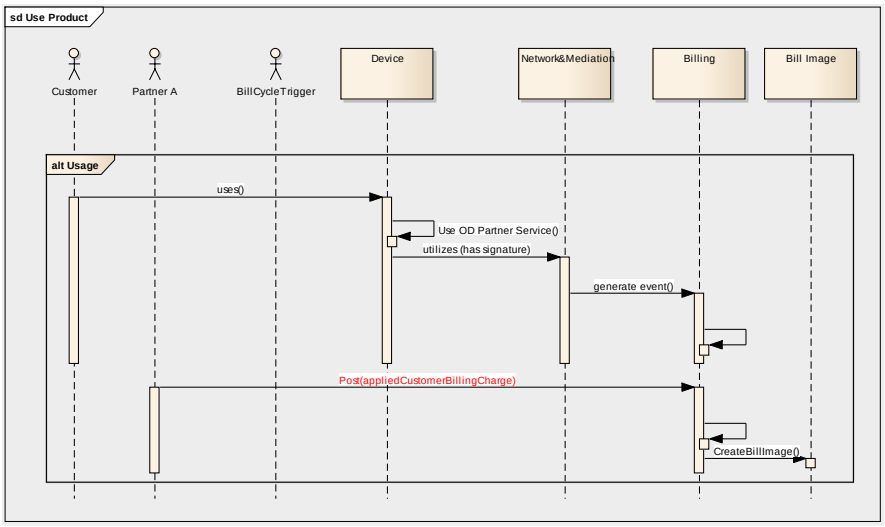

Figure 19 : Usage Event Use Case, subsequent Usage event and final consolidation on Bill Image.

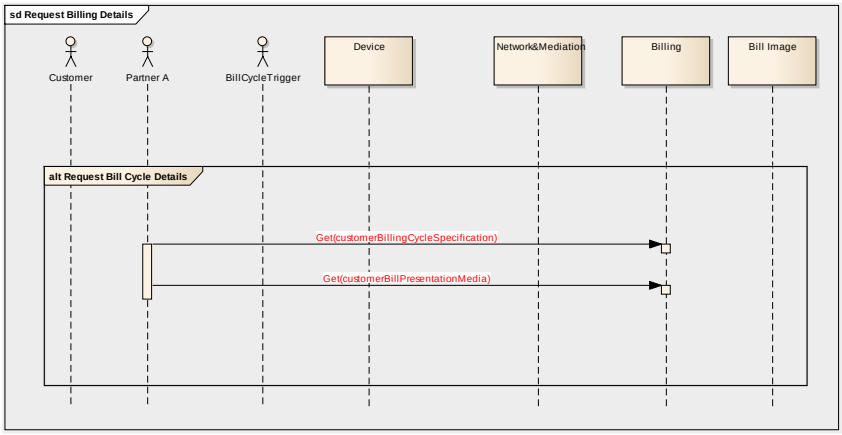

Figure 20 : Request Customer Billing Related Parameters (Either Bill Cycle or Bill presentation media)

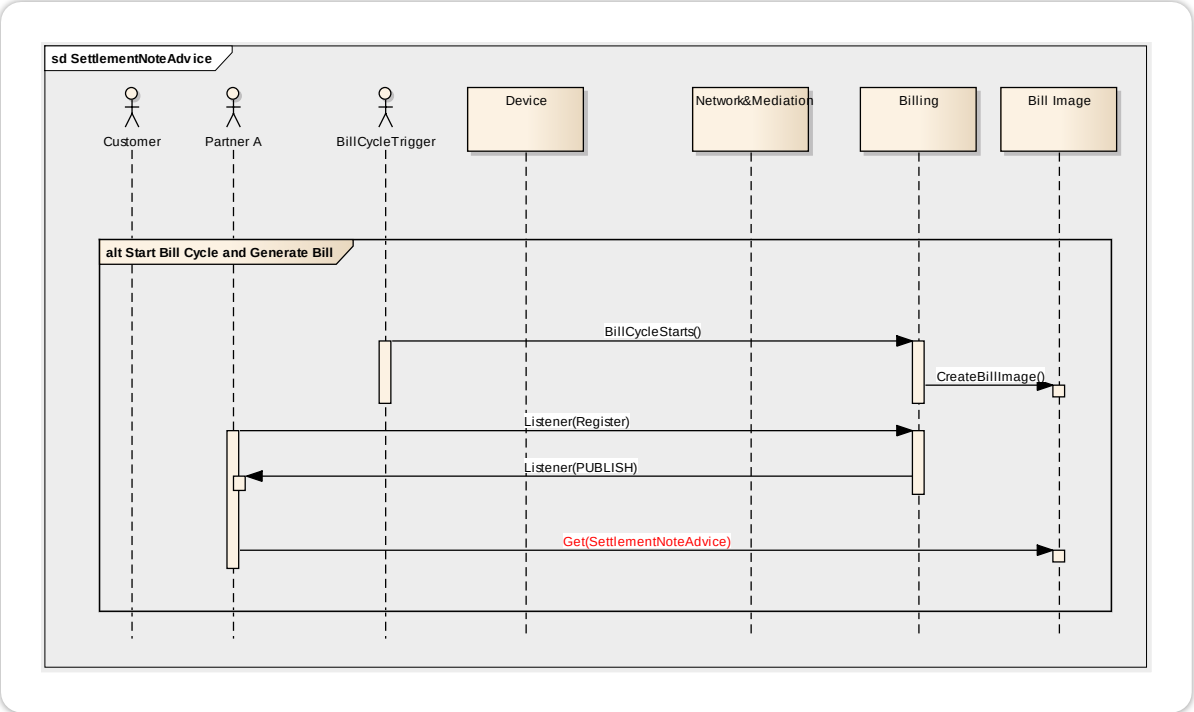

Figure 21 : Informing the partner about the availability of a Settlement Note Advice and subsequent actions by Partner to retrieve SettlementNoteAdvice

5. API description

This section provides a high level description of existing API

For each API, it describes:

- Its main features
- Resources managed
- Operation performed

5.1. Billing API

5.1.1. Overview

The Billing API provides standardized mechanisms for billing account, bill item and settlement note advice management either in B2B or B2B2C contexts.

It allows creation, update and retrieval of a billing account either in a B2B2C relationship context (creation of mass market customer billing account within a “Billing on Behalf of” process for example) or in a B2B context (creation of a billing account for a partner or B2B customer).

It allows also creation and query of bill items allowing partners or B2B customer to check their invoice.

In addition, it allows notification of settlement note advice to partners who can, then, query it.

5.1.2. Resources

Billing API manages the following Data resources:

- **Billing account**
    - A billing account is a detailed description of a customer’s bill structure
    - Main billing account attributes are its identifier and name, rating type (prepaid, postpaid), status, identifier of related customer account, billing cycle information, bill format and presentation media (post mail, email, web page), billing currency, billing account balance identifier, payment means (e.g. credit card) related party reference and role (Invoice responsible, payer, etc.)
- **Bill item** (aka Applied Customer Billing Charge in SID terminology)
    - A bill item is an amount, usually of money, for which a person or an organization is financially liable
    - Main bill item attributes are its identifier, creation date, description (additional data displayed on the bill), type of charge (recurring, one time or usage), currency code, taxes and amounts related information (tax included amount, tax excluded amount, tax rate), identifier and type of the concerned service
- **Settlement Note Advice**
    - The settlement is about transferring money between partners. The settlement is notified to the partner with a settlement note advice containing details in settlement lines.
    - Main settlement note advice attributes are its identifier, Creation date, description, due date, , taxes and amounts related information (tax included amount, tax excluded amount, tax rate, etc.), currency code, settlement method, receiver and issuer, settlement note Item related to a

product information (identifier, description, quantity, taxes and amounts related information, product information)

5.1.3. Operations

Billing API performs the following operations on the above mentioned resources

- **On billing account**
  - Retrieval of a billing account or a collection of billing accounts
  - Partial update of a billing account
  - Creation of a billing account
  - Notification of billing account state change
- **On Bill Item (Applied Customer Billing Charge)**
  - Creation of a bill item
  - Retrieval of a bill item or a collection of bill items
- **On Settlement note advice**
  - Retrieval of a settlement note advice or a collection of settlement note advices
  - Notification of a settlement note advice

5.2. Product ordering API

5.2.1. Overview

The Product Ordering API provides a standardized mechanism for placing a product order with all of the necessary order parameters. The API consists of a simple set of operations that interact with CRM/Order negotiation systems in a consistent manner. A product order is created based on a product offering that is defined in a catalog. The product offering identifies the product or set of products that are available to a customer, and includes characteristics such as pricing, product options and market.

The product order references the product offering and identifies any specific requests made by the customer.

5.2.2. Resources

Product Ordering API manages product order resource:

- A Product Order is a type of order which can be used to place an order between a customer and a service provider or between a service provider and a partner and vice versa,
- Main Product Order attributes are its identifier, state, priority category (mass market, Enterprise, etc.) related dates (start, completion, etc.), related billing account, related parties and order items
- Main Order Items (aka order lines) attributes are the ordered offering and product characteristics with the related action to be performed (e.g. add or delete the products), state, location information for delivery.

5.2.3. Operations

Product ordering API performs the following operations on Product Order:

- Retrieval of a product order or a collection of product orders depending on filter criteria
- Partial update of a product order (including updating rules)

- Creation of a product order (including default values and creation rules)
- Deletion of product order (for administration purposes)
- Notification of events on product order
    - Order creation
    - Order removal
    - Order state change
    - Order value change used to notify that any data in an order has just changed
    - Order information required used to notify that some data in the order need to be filled / are missing

5.3. Catalog management API

5.3.1. Overview

Catalog Management API provides a standardized solution for adding rapidly partners' products to an existing Catalog.

It brings the capability for Service Providers to directly feed partners systems with the technical description of the products they propose to them.

This reduces Catalog definition time and errors for offerings based on partners' products.

The Product Catalog API:

- Is used to communicate, expose, and mash-up product and service specifications into new product offerings in an efficient and easily understood manner
- Provides a straightforward and standardized mechanism for extracting data from catalogs, independently of platform, network or other implementation-specific technologies.
- Consists of a simple set of operations that interact with catalog management systems in a consistent way
- Provides a standardized way to do queries on Product Offering and Product Specifications

5.3.2. Resources

A Product Catalog is a collection of Product Offerings, intended for a specific Distribution Channel, enhanced with additional information such as SLA parameters, invoicing and shipping details. Each Product Offering in a Product Catalog combines pricing and availability information with Product Specifications that describe the relationships between Products, the Services used to realize the Products and the Resources they require.

The catalog management API allows:

- The management of the entire lifecycle of the catalog elements
- The lookup of the catalog elements during order capture

5.3.3. Operations

Catalog management API performs the following operations on catalog elements

- Retrieval of a catalog element or of a collection of catalog elements depending on filter criteria,
- Full update of catalog elements

- Partial update of catalog elements
- Creation of catalog elements
- Export of a catalog
- Import of a catalog
- Notification of changes on catalog elements

5.4. Trouble ticketing API

5.4.1. Overview

The Trouble ticketing API provides a standardized client interface to Trouble Ticket Management Systems for creating, tracking and managing trouble tickets among partners as a result of an issue or problem identified by a customer or another system. Examples of Trouble Ticket API clients include CRM applications, network management or fault management systems, or other trouble ticket management systems (e.g. B2B).

The API supports the ability to send requests to create a new trouble ticket specifying the nature and severity of the trouble as well as all necessary related information. The API also includes mechanisms to search for and update existing trouble tickets. Notifications are defined to provide information when a ticket has been updated, including status changes. A basic set of states of a trouble ticket has been specified to handle ticket lifecycle management.

5.4.2. Resources

Trouble ticketing API manages Trouble ticket resource:

- A trouble ticket represents a record used for reporting and managing the resolution of resource problems
- Main trouble ticket attributes are its description, severity, type, related dates (creation, target resolution, resolution), state, sub state and related information (change reason and change date), related parties (originator, owner, reviser, etc.) and notes

5.4.3. Operations

Trouble ticketing API performs the following operations on trouble ticket

- Retrieval of a trouble ticket or a collection of trouble tickets depending on filter criteria
- Full update of a trouble ticket
- Partial update of a trouble ticket
- Creation of a trouble ticket
- Notification of events on Trouble Ticket:
    - Ticket state change
    - Ticket change
    - Ticket clearance request
    - Information required

5.5. Service level agreement API

5.5.1. Overview

The SLA API provides a standardized interface for SLA life cycle Management (SLA Negotiation, SLA configuration SLA Activation/enforcement, SLA Operations, SLA violation / consequence handling, SLA reporting) between a Customer and a Service Provider which provides offers (product with attached SLA in its catalog) the customer can discover, browse, trigger and order.

It also will be also useful in a multi-partner environment where exchanging SLA is needed in order to allow rapid and efficient SLA life cycle management across partners' environment. From SLA perspective, duties and rights are assigned to each actor & associated roles mainly in the case where a service is composed of various components brought by different partners within federation or / and syndication models.

5.5.2. Resources

SLA Management API manages the following resources:

- SLA
    - Part of a business agreement between a Service Provider and a Customer, quantitatively specifying the service performance level the Service Provider commits to deliver. Other actors & roles can be involved such as SLA Auditor or SLA Integrator. SLA includes Service Level Specifications (SLS Parameters and Thresholds), as well as a description of measuring, reporting and violation handling processes. For the purpose of the specification, it can be expressed in terms of validity of period, related parties, and rules (metrics, reference value, tolerance, consequence …).
    - From a Customer perspective, this means that the end Customer provides Quality of Service requirements associated to its business applications to a Service Provider. The two parties negotiate the specific set of SLA parameters and parameter values that best serves them.
    - From the Service Provider perspective, each offered product or service can be provided with a standard Product SLA.
- SLA Violation
    - It represents any SLA failures observed through a metric threshold crossing restricted to what has been agreed in the SLAs for the given service (consequences, penalties, remedies...). SLA Violation is composed of KPIs, reported date, period, start Time, end Time. The Related Parties are represented in the same way as for "SLA" resource. This practical and operational view allows the related parties to react and perform an immediate and direct analysis of potential impacts of the violation.
    - There is also an "Attachment" which represents statistics, a dashboard or reporting data to be presented to the target parties, for deeper analysis purpose.

5.5.3. Operations

SLA API performs the following operations on SLA:

- Retrieval
    - all SLAs (with "SLA Provider", "SLA Customer" or "End User" role)
    - SLAs based on template
    - SLAs with specified ID – only one SLA is returned
- Creation of a SLA (planned)
- Full update of a SLA (planned)
- Partial update of a SLA (planned)

- Creation of a SLA violation (planned)
- Retrieval of a SLA violation
- Notification of SLA Violation Creation

5.6. Performance management API

5.6.1. Overview

The performance management API provides standardized mechanism for performance management such as creation, partial or full update and retrieval of the resources involved in performance management (Measurement Production Job, Measurement Collection Job, and Ad hoc Collection). It allows also notification of events related to performance.

5.6.2. Resources

Performance management API manages Measurement Production Job, Measurement Collection Job, and Ad hoc Collection resources.

A **Measurement Collection Job** is used to control the periodic collection of performance indicators, implemented as a sub-entity of the Performance Management Measurement Job.

Measurements production job is about controlling the generation of indicators, the equivalent of Performance Management Control in MTOSI. Production instructions are usually set in the set-up of a PM project. Changes are usually required for new versions of network elements or other kinds of network changes

Ad-hoc collection job is an on-demand collection method of Performance measurements results from a consuming application, requesting spontaneously (on-demand) a set of data for a set of network or service testing resources for a given temporal context. As the ad-hoc query is a single isolated request, its lifecycle is much simpler and isolated, having a request-response nature

5.6.3. Operations

Performance management API performs the following operations:

- Retrieval of Measurement Production Job, Measurement Collection Job and Ad hoc Collection
- Full update of Measurement Production Job, Measurement Collection Job and Ad hoc Collection
- Partial update of Measurement Production Job, Measurement Collection Job and Ad hoc Collection
- Creation of Measurement Production Job, Measurement Collection Job and Ad hoc Collection
- Deletion of Measurement Production Job, Measurement Collection Job, and Ad hoc Collection
- Notification of events:
    - Files ready
    - Files preparation error
    - Changes to ad hoc Collection, measurement Collection Job and measurement Production Job
    - Creation of measurement Collection Job and measurement Production Job
    - deletion of measurement Collection Job and measurement Production Job

5.7. Customer management API

5.7.1. Overview

The customer management API provides standardized mechanism for customer and customer account management, such as creation, update, retrieval, deletion and notification of events.

Customer can be a person, an organization or another service provider who buys products from an enterprise. Customer management API allows management of identification and financial information about him

5.7.2. Resources

Customer management API manages the following data resources:

- **Customer**
    - Customer represents a person or organization that buys products and services from the enterprise or receives free offers or services. Customers can also be other service providers who resell the enterprise's products, other service providers that lease the enterprise's resources for utilization by the other service provider's products and services, and so forth.
    - Customer resource contains information about the customer. Main attributes are its identifier, name, status and validity, description, characteristics, contact medium, related customer account, related party, customer credit profile information
- **Customer account**
    - Customer account represents a financial entity. It records all customer accounting events (payment and invoices amounts)
    - Customer account main attributes are its identifier, name, account type, status, description, credit limit, receivable balance, tax exemption, relationships, contacts, balance, payment plan and payment means

5.7.3. Operations

Customer management API performs the following operation on customer and customer account

- Retrieval of a customer or customer account or of a collection of them or depending on filter criteria
- Full update of a customer or customer account
- Partial update of a customer or customer account
- Creation of a customer or customer account
- Deletion of a customer or customer account
- Notification of events:
    - Customer creation
    - Customer update
    - Customer deletion
    - Customer account creation
    - Customer account update
    - Customer account delete

5.8. Party management API

5.8.1. Overview

The party management API provides standardized mechanism for party management such as creation, update, retrieval, deletion and notification of events.

Party can be an individual or an organization that has any kind of relation with the enterprise.

Party is created to record individual or organization information before the assignment of any role.

For example, within the context of a split billing mechanism, Party management API allows creation of the individual or organization that will play the role of 3rd party payer for a given offer and, then, allows retrieval or update of their information.

5.8.2. Resources

Party management API manages the following data resources:

- **Individual**
    - Individual represents a single human being (a man, woman or child). The individual can be a customer, an employee or any other person that the organization needs to store information about.
    - Main Individual attributes are its identifier, place of birth, country of birth, nationality, marital status, birth and death dates, disability, title, , names (given, family, middle, etc.), location, individual identification, related parties
- **Organization**
    - Organization represents a group of people identified by shared interests or purpose. Examples include business, department, and enterprise. Because of the complex nature of many businesses, both organizations and organization units are represented by the same data
    - Main organization attributes are its identifier, type, legal entity, date of creation and termination, trading name and type, other name, organization identifier, related parties and organization relationships (Parent and Child)

5.8.3. Operations

Party management API performs the following operations on individuals and organizations:

- Retrieval of an individual or organization or of a collection of them depending on filter criteria
- Full update of an individual or organization
- Partial update of an individual or organization
- Creation of an individual or organization
- Deletion of an individual or organization (for administration purposes)
- Notification of events:
    - Individual creation
    - Individual update
    - Individual deletion
    - Organization creation
    - Organization update
    - Organization delete

5.9. Usage API

5.9.1. Overview

The usage API provides standardized mechanism for usage management such as creation, update, retrieval, import and export of a collection of usages.

Usage API manages both rated and non-rated usage.

For example, Usage API allows a service provider:

- To retrieve usage generated by a partner service platform in order to rate it
- To provide rated usage to a partner for consumption follow up purposes

5.9.2. Resources

Usage API manages usage resource:

- A usage represents an event that is of billing system's interest and can have charges applied to it. It is comprised of characteristics, which define all attributes known for a particular type of usage
- Main usage attributes are its identifier, date, type, type related specification, characteristics (name and value), status, related parties and rating related information (date, amount, taxes, etc.)

5.9.3. Operations

Usage API performs the following operations on usage

- Retrieval of a usage or a collection of usage depending on filter criteria
- Partial update of a usage or of a collection of usage
- Creation of a usage
- Export of a collection of usages
- Import of a collection of usage

5.10. Product Inventory API

5.10.1. Overview

The Product Inventory API provides standardized mechanism for product inventory management such as creation, partial or full update and retrieval of the representation of a product in the inventory. It also allows the notification of events related to product lifecycle.

For example, product inventory API can be used to retrieve products owned by a customer or update the status of an installed product.

5.10.2. Resources

Product Inventory API manages product resource.

- Main product attributes are its identifier, name, description, status, related product offering, characteristics, specification, related billing account and parties, location, realized service and price information.

5.10.3. Operations

Product inventory API performs the following operations on product:

- Retrieval of the representation of a product inventory or of a collection of them depending on filter criteria
- Full update of the representation of a product
- Partial update of the representation of a product
- Creation of a new product entry in the inventory
- Deletion of a product in the inventory (for administration purposes)
- Notification of events:
    - Product creation
    - Product value change
    - Product status change
    - Product deletion
    - Product batch notification
    - Product synchronization

5.11. Service Inventory API

The Service Inventory API provides standardized mechanism for service inventory management such as creation, partial or full update and retrieval of the representation of a service in the inventory. It allows also notification of events related to service lifecycle.

For example, service inventory API can be used to retrieve services or update the status of a service.

5.11.1. Resources Model

Service Inventory API manages service resource:

- Main service attributes are its identifier, category, name, description, status, related services, characteristics, specification, related parties and supporting services & resources.

5.11.2. Operations

Service inventory API performs the following operations on service:

- Retrieval of the representation of a service inventory or of a collection of them depending on filter criteria
- Full update of the representation of a service
- Partial update of the representation of a service
- Creation of a new service entry in the inventory
- Deletion of a service in the inventory (for administration purposes)
- Notification of events:
    - Service creation
    - Service value change
    - Service status change
    - Service deletion

5.12. Resource Inventory API

The Resource Inventory API provides standardized mechanism for resource inventory management such as creation, partial or full update and retrieval of the representation of a resource in the inventory. It allows also notification of events related to resource lifecycle.

Logical and physical resources don't support any uniform operation as they are abstract resources.

Only concrete resources, such as FRE or shelf, support operations.

5.12.1. Resources Model

Resource Inventory API manages concrete resources:

- Main resource attributes are its identifier, type, name and alias, description, states, version, characteristics, specification, related logical resources (e.g. for FRE) and physical resources (e.g. for shelf).

5.12.2. Operations

Resource inventory API performs the following operations on resource:

- Retrieval of the representation of a physical resource (e.g. or shelf)
- Full update of the representation of a physical resource (e.g. or shelf)
- Partial update of the representation of a physical resource (e.g. or shelf)
- Creation of a new of a physical resource (e.g. or shelf)
- Deletion of a of a physical resource (e.g. or shelf)
- Notification of events:
    - Resource creation
    - Resource value change
    - Resource state change
    - Resource deletion

6. Gaps / Points to improve in a future release of this Guidebook

6.1. Suggested Short Term improvements

- Revisit and detail partner On-boarding process
- Extend business scenarios descriptions to fulfilment and resource ordering
- Elaborate / Detail Quick Start Building Blocks and Generic Patterns for API usage
- Provide a summarized visual overview of Digital Ecosystem
- Describe API related to Federate Authentication / Authorization / Identity / Profiling /Security
- Include API Ecosystem Requirements for Digital Services Reference architecture
- Additional improvements
    - Extend Customer Journey with 2) OPS view
    - Cover Functional & Management Perspectives? Lite
    - Draft a Management Capability Matrix (What is needed per BizM)
        - SQM (Resource Ordering) (Performance + x)
        - SMI

6.2. Long Term improvement suggestions

Provide a description of business scenario and the benefits of using API for the following use cases

- Smart Grid
- eHealth
- NFV

7. Resources description Appendix –

7.1. Billing API

Billing API manages the following Data resources:

- **Billing account**
  - A Billing Account is a detailed description of a customer's bill structure
  - Main billing Account attributes are :
    - Unique identifier of the billing account
    - short descriptive name
    - Rating type (prepaid, postpaid)
    - Status
    - Period of validity
    - Identifier of related customer account
    - Customer Billing Cycle Specification: A detailed description of when to initiate a billing cycle and the various sub steps of a billing cycle
    - Customer Bill Format: A detailed description of the way in which a customer's bill is presented
    - Customer Bill Presentation Media: A means of communicating a Customer Bill, supported by the associated bill format. For example, post mail, email, web page
    - Billing Currency
    - Identifier and name of Billing Account Balance
    - Payment Means which defines a specific mean of payment (e.g. direct debit with all details associated)
    - Related Party reference and role which identify the party playing a role within billing account (Invoice responsible, payer, etc.)
- **Bill item** (aka Applied Customer Billing Charge in SID terminology)
  - A bill item is an amount, usually of money, for which a person or an organization is financially liable.
  - Main attributes are
    - Unique identifier of the applied customer billing charge
    - Creation date of the applied customer billing charge
    - Description i.e. additional data to be displayed on the customer bill for this applied customer billing charge
    - Type of the applied customer billing charge (recurring, one time or usage)
    - Currency Code
    - All taxes included amount to be charged on the customer bill (expressed in the given currency)

- Tax excluded amount to be charged on the customer bill (expressed in the given currency)
- Tax Rate i.e. amount of money levied on the price of a product by a legislative body.
- Identifier and type of the concerned service

- **Settlement Note Advice**
  - The settlement is about transferring money receiving by a CSP to a partner. The settlement is notified to the partner with a settlement note advice containing details in settlement lines.
  - Main attributes are
    - unique identifier for the settlement note advice
    - Creation date of the settlement note advice
    - free description text
    - Date at which the due amount should have been paid to the partner
    - Date of the tax
    - Currency Code
    - All taxes excluded amount (expressed in the given currency)
    - All taxes included amount (expressed in the given currency)
    - Tax Item for each tax rate and tax type used in the settlement note
    - Settlement Method (the way in which the issuer provides the payment)
    - Receiver (Party who will receive the settlement note)
    - Issuer (Party who will issue the settlement note)
    - Issuer and receiver Tax Registration number
    - image of the settlement note
    - Settlement Note Item which provides information about amount settled for a product identified by its catalog reference
      - Unique number assigned to the settlement note item
      - Unique identifier of the settlement note item
      - A free description text for the settlement note item
      - Quantity
      - Tax excluded unit price to be applied on the given quantity
      - Tax excluded amount equal to quantity * tax Excluded Unit Price
      - All taxes included amount equal to tax Excluded Amount + tax Amount
      - Identification number assigned to the product
      - Name of the product

7.2. Product ordering API

Product Ordering API manages the following Data resources:

- **Product order**
    - Main Product Order attributes are
        - Id: ID created on repository side (OM system)
        - Href: Hyperlink to access the order
        - External ID: ID given by the consumer and only understandable by him (to facilitate his searches afterwards)
        - State: State of the order
        - Priority: A way that can be used by consumers to prioritize orders in OM system (from 0 to 4 : 0 is the highest priority, and 4 the lowest)
        - Category: Used to categorize the order from a business perspective that can be useful for the OM system (e.g. “enterprise”, “residential”, ...)
        - Order Date: Date when the order was created
        - Requested Start Date: Order start date wished by the requestor
        - Completion Date: Date when the order was completed
        - Requested Completion Date: Requested delivery date from the requestor perspective
        - Expected Completion Date: Expected delivery date amended by the provider
        - Notification Contact: Contact attached to the order to send back information regarding this order
        - Note: Extra-information about the order (e.g. useful to add extra delivery information that could be useful for a human process: a digicode access to a building …)
        - Billing Account: Billing Account to use to bill the ordered products
        - Related Party: Parties which are involved in this order and the role they are playing
        - Order Item: List of order items that have to be treated
- **Order item**
    - Main Product Order attributes are
        - Id: Identifier of the line item (generally it is a sequence number 01, 02, 03, …)
        - Action Can be “add” / “modify” / “no_change”/ “delete”
        - State State of the order item
        - Place: Used to defined a place useful for the order item (for example a delivery geographical place)
        - Appointment: Used to precise that an appointment was set up with a related party for this order item
        - Product Offering :Ordered offering (pricing, default values, etc. are fetched by the OM directly from the catalog)
            - It may be a bundle Product Offering, in this case it will contain the list of bundled offers that are ordered
            - Product: Configure the product characteristics (only configurable characteristics and necessary only if a non-default value is selected) and/or identify the product that needs to be modified/deleted. Product attributes are:
                - Id: Identifier of the owned product (useful for delete or modify command)

- Href: Reference to the owned product (useful for delete or modify command)
- Product Characteristics: Characteristics of the product to instantiate or to modify
- Product Relationship: Linked products to the one instantiated, it can be :
  - "bundled" if the product is a bundle and you want to describe the "bundled" products inside this bundle
  - "reliesOn" if the product needs another already owned product to rely on (e.g. an option on an already owned mobile access product)
  - "targets" or "isTargeted" (depending on the way of expressing the link) for any other kind of links that may be useful
  - Related Party: Party linked at the product level (it may be a User for example)

7.3. Catalog management API

Catalog management API manages the following Data resources:

- **Attachment** which describes a product through video, pictures... with the following attributes
  - Description: Description of the attachment
  - Href: Reference of the attachment
  - Id: Unique identifier of the attachment
  - Type Attachment type such as video, picture
  - Uri:Uniform Resource Identifier, describes the mechanism used to access the resource, the specific computer that the resource is housed in, the specific name of the resource (file name) on the computer
  - url: Uniform Resource Locator, is a web page address (a subset of URI)
- **Bundled Product Offering**: A type of Product Offering that belongs to a grouping of Product Offerings made available to the market. It inherits of all attributes of Product Offering.
- **Bundled Product Specification**: A type of Product Specification that belongs to a grouping of Product Specifications made available to the market. It inherits of all attributes of Product Specification.
- **Catalog**: A Product Catalog is a collection of Product Offerings, intended for a specific Distribution Channel, enhanced with additional information such as SLA parameters, invoicing and shipping details. Catalog attributes are:
  - Category: The category resource is used to group product offerings, service and resource candidates in logical containers. Categories can contain other categories and/or product offerings, resource or service candidates and have the following attributes:
    - Description: Description of the category
    - Href: Reference of the catalog
    - Id: Unique identifier of the category
    - Is Root : If true, this Boolean indicates that the category is a root of categories

        - Last Update: Date and time of the last update
        - Lifecycle Status: Used to indicate the current lifecycle status
        - Name: Name of the category
        - Parent Id: Unique identifier of the parent category
        - Valid For: The period for which the category is valid
        - Version: Category version
    - Href: Reference of the catalog
    - Id: Unique identifier of the catalog
    - Last Update: Date and time of the last update
    - Lifecycle Status: Used to indicate the current lifecycle status
    - Name: Name of the catalog
    - Type: Indicates if the catalog is a product, service or resource catalog
    - Valid For: The period for which the catalog is valid
    - Version: Catalog version
- **Channel**: Defines the channel for selling product offerings with the following attributes
    - Href: Reference of the channel
    - Id: Unique identifier of the channel
    - Name: Name of the channel
- **Place**: Defines the places where the product offerings are sold with the following attributes:
    - Href: Reference of the channel
    - Id: Unique identifier of the channel
    - Name: Name of the channel
- **Price**: Provides all amounts (tax included, duty free, tax rate), used currency and percentage to apply for Prod Offer Price Alteration with the following attributes:
    - Currency Code : A string used as a code for specifying the currency associated to the given amounts. The ISO4217 norm uses 3 letters to define the currency (for example USD for US dollar or EUR for Euro)
    - Duty Free Amount: All taxes excluded amount (expressed in the given currency)
    - Percentage: Percentage to apply for Prod Offer Price Alteration
    - Tax Included Amount: All taxes included amount (expressed in the given currency)
    - Price Type: Indicates the price type: recurring, one time, usage
- **Product Offering**: The Product Offering resource represents entities that are orderable from the provider of the catalog, this resource includes pricing information.
    - Description: Description of the product Offering
    - Href: Reference of the Product Offering
    - Id: Unique identifier of the product Offering
    - Is Bundle: is Bundle determines whether a product Offering represents a single product offering (false), or a bundle of product Offerings (true).
    - Last Update: Date and time of the last update

- Lifecycle Status: Used to indicate the current lifecycle status
- Name: Name of the product Offering
- validFor: The period for which the product Offering is valid
- version: Product Offering version

- **Product Offering Price**: is based on both the basic cost to develop and produce products and the enterprise's policy on revenue targets. This price may be further revised through discounting (prod Offer Price Alteration):
  - Description: Description of the product Offering Price
  - Href: Reference of the Product Offering Price
  - Id: Unique identifier of the product Offering Price
  - Name: Name of the product Offering Price
  - Price Type: Indicates the price type: recurring, one time, usage
  - Recurring Charge Period: Could be month, week…
  - Unit of Measure: Could be minutes, GB…
  - Valid For: The period for which the product Offering Price is valid
  - Version: Product Offering Price version
- **Prod Offer Price Alteration**: is an amount, usually of money, that modifies a price charged for a product Offering.
  - Application Duration: Duration during which the prod Offer Price Alteration applies on the product Offering
  - Description: Description of the prod Offer Price Alteration
  - Href: Reference of the prod Offer Price Alteration
  - Id: Unique identifier of the prod Offer Price Alteration
  - Name: Name of the prod Offer Price Alteration
  - Price Condition: Condition that triggers the price application
  - Price Type: Indicates the price type: recurring, one time, usage
  - Recurring Charge Period: Could be month, week…
  - Unit of Measure: Could be minutes, GB…
  - Valid For: The period for which the prod Offer Price Alteration is valid
- **Product Offering Term** is a condition under which a Product Offering is made available to Customers. For instance, a product Offering can be offered with multiple commitment periods.
  - Description: Description of the product Offering Term
  - Duration: Duration of the product Offering Term
  - Name: Name of the product Offering Term
  - Valid For: The period for which the product Offering term is valid
- **Product Spec Characteristic** is a characteristic quality or distinctive feature of a Product Specification. The characteristic can be taken on a discrete value, such as color, can take on a range of values or can be derived from a formula:
  - Configurable: If true, the Boolean indicates that the product Spec Characteristic is configurable

- Description: A narrative that explains in detail what the product Spec Characteristic is
- Href: Reference of the product Spec Characteristic
- Id: Unique identifier of the product Spec Characteristic
- Name: Name of the product Spec Characteristic
- Value Type: A kind of value that the characteristic can take on, such as numeric, text and so forth
- Valid For: The period for which the product Spec Characteristic is valid

- **Product Spec Char Relationship**: An aggregation, migration, substitution, dependency or exclusivity relationship between/among product Spec Characteristics.
  - Href: Reference of the product Specification
  - Valid For: The period for which the product Spec Char Relationship is valid
  - Type: Type of relationship such as aggregation, migration, substitution, dependency, exclusivity
- **Product Spec Characteristic Value**: A number or text that can be assigned to a Product Spec Characteristic.
  - Default: Indicates if the value is the default value for a characteristic
  - Unit Of Measure: Could be minutes, GB…
  - Valid For: The period of time for which a value is applicable
  - Value: A discrete value that the characteristic can take on
  - Value From: The low range value that a characteristic can take on
  - Value To: The upper range value that a characteristic can take on
  - Value Type: A kind of value that the characteristic can take on, such as numeric, text, and so forth
- **Product Specification**: A detailed description of a tangible or intangible object made available externally in the form of a Product Offering to customers or other parties playing a party role.
  - Brand: The manufacturer or trademark of the specification
  - Description: A narrative that explains in detail what the product specification is
  - Href: Reference of the product specification
  - Id: Unique identifier of the product specification
  - Is Bundle: is Bundle determines whether a product Specification represents a single product Specification (false), or a bundle of product Specification (true).
  - Last Update: Date and time of the last update
  - Lifecycle Status: Used to indicate the current lifecycle status
  - Name: Name of the product specification
  - Product Number: An identification number assigned to uniquely identify the specification
  - Valid For: The period for which the product specification is valid
  - Version Product: specification version
- **Product Specification Relationship** is a migration, substitution, dependency or exclusivity relationship between/among product specifications.
  - Href: Reference of the product Specification

- Valid For: The period for which the product Specification Relationship is valid
- Type: Type of relationship such as migration, substitution, dependency, exclusivity

- **Related Party** Defines party or party Role linked to a specific entity.
  - Href: Reference of the related Party, could be a party reference or a party Role reference
  - Name: Name of the related party
  - Role: Role of the related party
  - Valid For: The period for which the related party is linked to the entity
- **Required Resource Specification** is the Resource Specification required to realize a resource Specification.
  - Href: Reference of the resource Specification
  - Name: Name of the required Resource Specification
  - Valid For: The period for which the Required Resource Specification is valid
- **Required Service Specification** is the Service Specification required to realize a Service Specification.
  - Href: Reference of the service Specification
  - Name: Name of the required Service Specification
  - Valid For: The period for which the Required Service Specification is valid
- **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.
  - Description: a narrative that explains in detail what the resource candidate is
  - Href: Reference of the resource candidate
  - Id: Unique identifier of the resource candidate
  - Last Update: Date and time of the last update
  - Lifecycle Status: Used to indicate the current lifecycle status
  - Name: Name of the resource candidate
  - Valid For: The period for which the resource candidate is valid
  - Version: Resource candidate version
- **Resource Spec Characteristic** defines the characteristic features of a Resource Specification. Every Resource Specification has a variety of important attributes, methods, constraints, and relationships, which distinguish that Resource Specification from other Resource Specifications.
  - Configurable: If true, the Boolean indicates that the resource Spec Characteristic is configurable
  - Description: a narrative that explains in detail what the resource Spec Characteristic is
  - Href: Reference of the resource Spec Characteristic
  - Id: Unique identifier of the resource Spec Characteristic
  - Name: Name of the resource Spec Characteristic
  - Value Type: A kind of value that the characteristic can take on, such as numeric, text and so forth
  - Valid For: The period for which the resource Spec Characteristic is valid

- **Resource Spec Characteristic Value** defines the characteristic features of a Resource. Every Resource has a variety of important attributes and relationships, which distinguish that Resource from other Resources.
    - Default: Indicates if the value is the default value for a characteristic
    - Unit Of Measure: Could be minutes, GB…
    - Valid For: The period of time for which a value is applicable
    - Value: A discrete value that the characteristic can take on
    - Value From: The low range value that a characteristic can take on
    - Value To: The upper range value that a characteristic can take on
    - Value Type: A kind of value that the characteristic can take on, such as numeric, text, and so forth
- **Resource Specification** is an abstract base class for representing a generic means for implementing a particular type of Resource.
    - Brand: The manufacturer or trademark of the specification
    - Description: A narrative that explains in detail what the resource specification is
    - Href: Reference of the resource specification
    - Id: Unique identifier of the resource specification
    - Last Update: Date and time of the last update
    - Lifecycle Status: Used to indicate the current lifecycle status
    - Name: Name of the resource specification
    - Valid For: The period for which the resource specification is valid
    - Version: Resource specification version
- **Service Candidate**: Is an entity that makes a Service Specification available to a catalog.
    - Description: A narrative that explains in detail what the service candidate is
    - Href: Reference of the service candidate
    - Id: Unique identifier of the service candidate
    - Last Update: Date and time of the last update
    - Lifecycle Status: Used to indicate the current lifecycle status
    - name Name of the service candidate
    - Valid For: The period for which the service candidate is valid
    - Version: Service candidate version
- **Service Spec Characteristic** is an abstract base class that represents the key features of this Service Specification.
    - Configurable: If true, the Boolean indicates that the service Spec Characteristic is configurable
    - Description: A narrative that explains in detail what the service Spec Characteristic is
    - Href: Reference of the service Spec Characteristic
    - Id: Unique identifier of the service Spec Characteristic
    - Name: Name of the service Spec Characteristic

  - Value Type: A kind of value that the characteristic can take on, such as numeric, text and so forth
  - Valid For: The period for which the service Spec Characteristic is valid
- **Service Spec Characteristic Value**: Is used to define a set of attributes, each of which can be assigned to a corresponding set of attributes in a Service Spec Characteristic object.
  - Default: Indicates if the value is the default value for a characteristic
  - Unit Of Measure Could be minutes, GB…
  - Valid For: The period of time for which a value is applicable
  - Value: A discrete value that the characteristic can take on
  - Value From: The low range value that a characteristic can take on
  - Value To: The upper range value that a characteristic can take on
  - Value Type: A kind of value that the characteristic can take on, such as numeric, text, and so forth
- **Service Specification** is an abstract base class for representing a generic means for implementing a particular type of Service.
  - Brand: The manufacturer or trademark of the specification
  - Description: A narrative that explains in detail what the service specification is
  - Href: Reference of the service specification
  - Id: Unique identifier of the service specification
  - Last Update: Date and time of the last update
  - Lifecycle Status: Used to indicate the current lifecycle status
  - Name: Name of the service specification
  - Valid For: the period for which the service specification is valid
  - Version: Service specification version
- **Service Spec Relationship** allows for Service Specifications to contain other Service Specifications.
  - Href: Reference of the service Specification
  - Valid For: The period for which the service Spec Relationship is valid
  - Type: Type of relationship such as migration, substitution, dependency, exclusivity

7.4. Trouble ticketing API

Trouble ticketing API manages trouble ticket Data resource.

Main trouble ticket attributes are:

- Id: unique identifier of the trouble ticket
- Correlation Id: Additional trouble ticket identifier coming from an external system
- Description: description of the trouble
- Severity: severity of the trouble. It can be for example: minor, major, critical
- Type: Type of trouble ticket

- Creation Date: The date on which the trouble was discovered
- Target Resolution Date: Foreseen trouble resolution date
- Status: Current status of the Trouble Ticket
- Sub Status: Current sub status of the Trouble Ticket
- Status Change Reason: The reason of state change
- Status Change Date: The date of state change
- Resolution Date: The date on which the service was brought back to its original condition
- Related Party: Party playing a role within trouble ticket with their role and reference
- Related Object: Objects linked with trouble ticket with their involvement and reference
- Note: Extra-information about the trouble ticket including date, author and text of the note

7.5. Service level agreement API

Service level agreement API manages:

- Service Level Agreement resource
- Service Level Agreement violation resource

Main attributes of Service Level Agreement are:

- Id: id of the Service level Agreement (SLA)
- Name: Name of the Service level Agreement (SLA)
- Description: Description of the Service level Agreement (SLA)
- Version : Version of the Service level Agreement (SLA)
- Validity Period: Period where the clauses of the SLA are in force
- Start time: Time where the clauses of the SLA are executed
- End time: Time where the clauses of the SLA are finished
- Template: SLA Template characteristics
    - Href: Reference of the Template
    - Name: Name of the template
    - Description: Description of the template
    - Related Party: Parties engaged in the SLA (SLA provider, SLA consumer, …) with Role and reference
- State: Service Level Agreement state
- Approved: Indicates if the service level agreement is approved or not (True or false)
- Rules: Common pattern or Template for the Service Level Specification
    - Id: Unique identifier of the metric
    - Metric: Reference of metric stored in the Service Provider “Metrics Library”
    - Unit: Unit of measure of metric
    - Reference Value: Reference value of metric

- Operator: Operator used when calculating the rule
- Tolerance Allowable variation of metric
- Consequence: defines the action to be applied as a result of a threshold crossing

Main attributes of Service Level Agreement violation are:

- Id: The id of the SLA Violation)
- SLA : of the Service level Agreement description and reference
- Related Party: Parties engaged in the SLA (SLA provider, SLA consumer, …) with Role and reference
- Violation: A discrepancy identified by applying rules to Service Level Agreement related entities
  - Rule: Common pattern or Template for the Service Level Specification with description and reference
  - Unit: Unit of measure of metric
  - Reference Value: Reference value of metric
  - Operator: Operator used when calculating the rule
  - Actual Value: Actual value of the metric
  - Tolerance: Allowable variation of metric
  - Violation Average: TBD
  - Comment: Comment about violation
  - Consequence: defines the action to be applied as a result of a threshold crossing
  - Attachments: represents statistics, a dashboard or reporting data to be presented to the target parties with description and reference

7.6. Performance management API

To be completed

7.7. Customer management API

Customer management API manages:

- Customer resource
- Customer account resource

Main attributes of Customer are:

- Id: Unique identifier of the customer
- Name: Displayable name
- Status: Used to track the lifecycle status, e.g. existing, prospective or former customers
- Description: Detailed description of the customer
- Valid For: Extension to manage start Date and end Date of customer states

- Customer Rank: Relative importance of this customer compared to other customers
- Characteristics: Characteristics of the customer (only the one valid at the moment : no history is managed on characteristics)
- Contact Medium: Describes the contact mediums that could be used to contact the customer (emails, phone numbers, postal addresses)
- Customer Account: Accounts of the customer
- Related Party: Party linked to this customer: the link used can be an URI to an organization or an individual depending on the type of customer.
- Customer Credit Profile: Credit profile for the customer (containing credit scoring …). By default only the current credit profile is retrieved. It can be used as a list to give the customer credit profiles history; the first one in the list will be the current one.
    - Credit Profile Date: Date when the profile was established
    - Credit Risk Rating: This is an integer whose value is used to rate the risk of this Customer paying late or defaulting versus paying on time.
    - Credit Score: A measure of a person or organization creditworthiness calculated on the basis of a combination of factors such as their income and credit history.
    - Valid For: The period for which the profile is valid.

Main attributes of Customer Account are:

- Id: Unique identifier for the customer account
- Last Modified: date of last modification of customer account
- Name: The name of the account.
- Account Type: A categorization of an account, such as individual, joint, and so forth, whose instances share some of the same characteristics.
- Status: The condition of the account, such as “due”, “paid”, “in arrears”, “in collection”.
- Description: Detailed description of the customer
- Credit Limit: The maximum amount of money that may be charged on an account.
- Pin: A multi digit personal identification number that is used by a Customer to gain access to a Customer Account
- Receivable Balance: Overall receivable balance for the customer account
- Customer Account Tax Exemption: Proof of freedom from taxes imposed by a taxing jurisdiction.
    - Issuing Jurisdiction: name of the taxing jurisdiction for which taxes are exempt
    - Certificate Number: identifier of a document that shows proof of exemption from taxes for the taxing jurisdiction.
    - Valid For: period for which the exemption is valid.
    - Reason: Reason of the tax exemption
- Customer Account Relationship: Significant connection between Customer Accounts.
    - Relationship Type: Type of relationship
    - Valid For: Validity period of that relationship
    - Customer Account: The target linked customer account
- Contact :

- Contact Type: Type of contact: primary, secondary …
- Valid For: Validity period of that contact
- Contact Medium: The contact to access
- Contact Name: A displayable name for the contact
- Party Role Type: precise what kind of Party Role type is linked to this contact (a customer account manager, …)
- Related Party: A link to the party whose contact is represented here (only if possible and useful)

- Customer: A link to the customer
- Customer Account Balance: Balances linked to the customer account
  - Id: Identifier of the customer account balance
  - Type: Deposit balance, Disputed balance, Loyalty balance, Receivable balance
  - Amount: Balance amount
  - Status: Due, Paid …
  - Valid For: Balance validity period
- Payment Plan: Defines a plan for payment (when a customer wants to spread his payments)
  - Id: Identifier of the payment plan
  - Status: Status of the payment plan (effective, ineffective).
  - Type: Type of payment plan
  - Priority: Priority of the payment plan
  - Amount: Amount paid.
  - Payment Frequency: Monthly, Bimonthly …
  - Number Of Payments: Number of payments used to spread the global payment.
  - Valid For: Validity period of the payment plan.
  - Payment Means: For each payment plan a payment method has to be specified

7.8. Party management API

Customer management API manages:

- Individual resource
- organization resource

Main attributes of individual are:

- Id: Unique identifier for the party
- Gender: Gender
- Place Of Birth: An hyperlink to the place of birth
- Country Of Birth: The country of birth

- Nationality: Nationality
- Marital Status: Marital status (married, divorced, widow, …)
- Birth date: Birth date
- Death Date: Death date
- Characteristic: Allows to describes individual hobbies, center of interests …
- Disability: List of disabilities (such as blind, motor-handicapped, …)
- Title: Useful for titles (aristocratic, social …): Pr, Dr, Sir …
- Given Name: First name
- Family Name: Last name
- Middle Name: Middle names
- Full Name: Full name flatten (first, middle, and last names)
- Formatted Name: A formatted name useful for specific contexts (Chinese, Japanese, Korean, …)
- Other Name: To keep track other names (for example the old name of a woman before marriage or an artist name)
- Location: Temporary current location of the party (may be used if the party has approved its sharing)
- Individual Identification: Identifiers used to identify a person (passport, national identity card, driver's license, social security number, birth certificate)
- External Reference: External reference to manage touch points to external OTT identifiers for the person (Facebook, google+ …). In addition to the SID Model.
- Related Party: List of party Roles (attributes are expanded here) with the association of Party Role to party. In this construct, Party Role is not a managed resource.
- Contact Medium: Describes the contact mediums that could be used to contact the customer (emails, phone numbers, postal addresses)

Main attributes of organization are:

- Id: Unique identifier for the party
- Type: Type of Organization (Company, …)
- Is Legal Entity: To tag if the organization is a legal entity known by national referential
- Exists During: Details the establishment of the organization and its cessation
- Trading Name: The name that the organization (unit) trades under
- Name Type: "Co.", "Inc.", "Ltd.", "Pty Ltd." , "Plc.", "Gmbh"
- Other Name: May be used to put former names
- Characteristic: Allows to describe specific characteristics of organization
- Organization Identification: Used to identify company in a country or internationally
- External Reference: External reference to manage touch points to external OTT identifiers for the person (Facebook, google+,) In addition to the SID Model.
- Related Party: List of partyRoles (attributes are expanded here) with the association of PartyRole to party. In this construct, PartyRole is not a managed resource. In addition to the SID Model.
- Organization Relationships (Parent and Child): Links between organizations: useful to describe an organization structure between headquarters, affiliates, In addition to the SID Model.

- Contact Medium : Describes the contact mediums that could be used to contact the customer (emails, phone numbers, postal addresses)

7.9. Usage API

Usage API manages:

- Usage resource
- Usage specification resource

Main attributes of usage are:

- Id: Usage unique identifier
- Date: Date of usage
- Type: Type of usage
- Description: Description of usage
- Usage Specification: A detailed description of a usage event that are of billing system's interest and can have charges applied to it. It is comprised of characteristics, which define all attributes known for a particular type of usage
- Usage Specification Characteristic: A detailed description of an attribute that defines a particular type of usage, described by its name, category, type, presence and a set of allowed values
- Status: Status of usage
- Related Party: Party linked to this usage with its role and reference
- Rated product Usage: Describes the characteristics of rated usage
    - Rating Date: Date of usage rating
    - Usage Rating Tag TBD
    - Is Billed: Boolean indicating if usage have been billed or not
    - Rating Amount Type: Type of amount
    - Tax Included Rating Amount: All taxes included rated amount
    - Tax Excluded Rating Amount: All taxes excluded rated amount
    - Tax Rate: Tax rate
    - Is Tax Exempt: Indicates if the rated amount is exempt of tax
    - Offer Tariff Type: Type of tariff applied
    - Bucket Value Converted In Amount: Monetary value of bucket
    - Currency Code: A string used as a code for specifying the currency associated to the given amounts. The ISO4217 norm uses 3 letters to define the currency (for example USD for US dollar or EUR for Euro)
    - Product Ref: Reference of product specification

Main attributes of usage specification are:

- Id : Usage specification unique identifier
- Name: The name of the usage specification.

- Description: A narrative that explains in detail what the product specification is.
- Valid For: Extension to manage start Date and end Date of usage Specification
- Usage Spec Characteristic: A detailed description of an attribute that defines a particular type of usage, described by its name, category, type, presence and a set of allowed values
    - Name: The name of the usage characteristic.
    - Description: A narrative that explains the usage characteristic.
    - Configurable Boolean indicating if usage Spec Characteristic is configurable or not
    - Usage Spec Characteristic Value: A value that can be assigned to a Usage Spec Characteristic.
        - Value Type: A kind of value that the characteristic can take on, such as numeric, text, and so forth.
        - Default: Indicates if the value is the default value for a characteristic
        - Value: A discrete value that the characteristic can take on.
        - Value From: The low range value that a characteristic can take on
        - Value To: The upper range value that a characteristic can take on.

7.10. Product Inventory API

Product inventory API manages Product resource:

Main attributes of Product are:

- Id: is the ID created for the product
- Name: is the name of the product. It could be the same as the name of the Product Offering
- Description: is the description of the product. It could be copied from the description of the Product Offering.
- Status: Is the status of the product.
- isBundle: Boolean indicating if the product is a bundle
- is Customer Visible: is a Boolean indicating if the product is visible by customer
- product Serial Number: is the serial number for the product. This is typically applicable to tangible products e.g. Broadband Router.
- Start Date: is the date from which the product starts
- Order Date: is the date when the product was ordered.
- Termination Date: is the date when the product was terminated. Not applicable to active products.
- Product Offering: is a link to the product offering.
- Product Specification: is a link to the product specification. It may be useful to manage links directly to the Product Specification, when the product is a non-bundle one you may not manage the Product Offering link with this Product.
- Product Characteristics: is a list of name value pairs listing the various relevant characteristics of the instantiated product.

- Product Relationships: is a collection of a number of types of relationships between the products e.g. bundle. The value of the 'type' attribute should describe the direction of the relationship. E.g. "isContainedIn"
- Billing Account: is a link to the billing account on which the product is billed. This is optional because in some cases the parent is billed and not all the children.
- Related Parties: is a link to collection of parties. Each party will contain a role and a link.
- Agreement Item: is a link to the agreement item for the product. Agreement item will contain the terms and conditions for the product including the product term.
- Place: is a link to the place where the product is provided.
- Realizing Service: is a link to the service that realizes the product.
- Realizing Resource: is a link to the resource that realizes the product.
- Product Price: is a collection of all applicable prices for the product, it's an override of the default one specified in the Product Offering.

7.11. Service Inventory API

Service inventory API manages Service resource:

Main attributes of service are:

- Id: is the ID created for the service
- Href: reference of the service
- Category: indicate if it is a customer facing or resource facing service
- Name: is the name of the service.
- Description: is the description of the service.
- Is Service Enabled: signifies if service is enabled for use.
- Has Started signifies that this service has been started.
- Start Mode enumerated integer that indicates how the Service is started.
- Is Stateful: means that this service can be changed without affecting any other services.
- Status: is the lifecycle status of the product. Possible values are :
    - "operational"
    - "degraded"
    - "inViolation"
    - "inTest"
    - "beingDeployed"
    - "failed"
    - "terminated"
    - "supporting CFS" links are not captured through the states (RFS), same solution for "billable" (CFS)
- Service Specification: is a link to the service specification.

- Service Characteristics : is a list of name value pairs that define the service characteristics.
- Service Relationship: Describes links with services of the same category (useful for bundled services)
- Supporting Service: A collection of services that support this service (links between CFS -> RFS)
- Supporting Resource: A collection of resources that support this service (links between RFS -> Resources)
- Related Party: A link to collection of parties. Each party will contain a role and a link.

7.12. Resource Inventory API

To be completed

7.13. Service Management API

The Diagram below extends the standard overview to show how service Management interacts with other operational process (Service Management highlighted in Green)

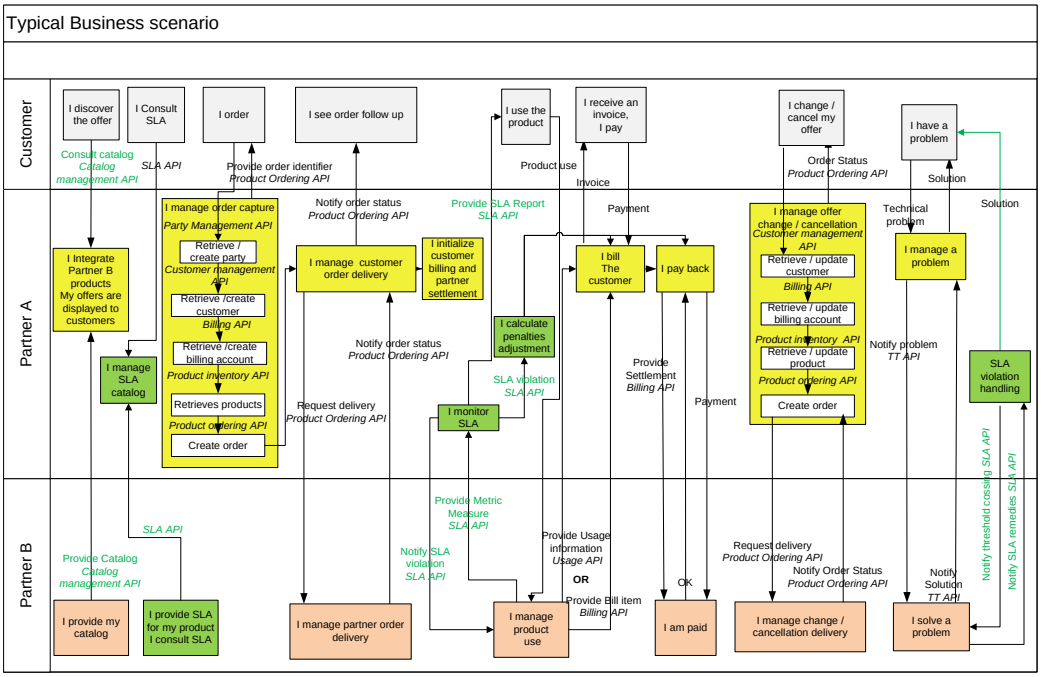

Figure 22 : Operational Process interworking with Service Management API

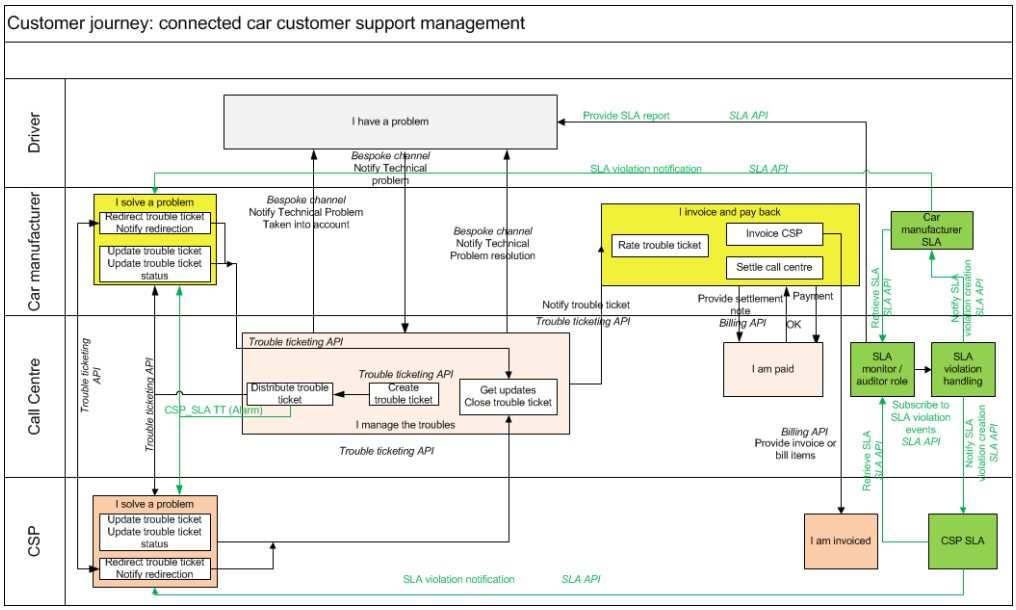

Figure 23 : Service Management Relation to T2R for the connected Car Scenario.

8. Administrative Appendix

This Appendix provides additional background material about the TM Forum and this document. In general, sections may be included or omitted as desired; however, a Document History must always be included.

8.1. About this document

This is a TM Forum Guidebook. The guidebook format is used when:

- The document lays out a 'core' part of TM Forum's approach to automating business processes. Such guidebooks would include the Telecom Operations Map and the Technology Integration Map, but not the detailed specifications that are developed in support of the approach.
- Information about TM Forum policy, or goals or programs is provided, such as the Strategic Plan or Operating Plan.
- Information about the marketplace is provided, as in the report on the size of the OSS market.

8.2. Document History

8.2.1. Version History

Version Number	Date Modified	Modified by:	Description of changes
0.1	11 July 2014	Jean-Luc Tymen	Initial version for Team Action Week contribution
0.2	25 July 2014	Johannes Minnaar	Additional Input from before and during TAW
0.3	14 Oct 2014	Jean-Luc Tymen	Merge of updates From J. Minnaar and JL. Tymen
0.3.1	12 Mar 2015	Alicja Kawecki	Updated cover, footer and Notice to reflect TM Forum Approved status

8.2.2. Release History

Release Number	Date Modified	Modified by:	Description of changes
R1.0	11 July 2014	Jean-Luc Tymen	Initial version for Team Action Week contribution
R1.1	10 Oct 2014	Johannes Minnaar	Updated Draft with contributions
R1.2	14 Oct 2014	Jean-Luc Tymen	Merge of updates From J. Minnaar and JL. Tymen

8.3. Company Contact Details

Company	Team Member Representative
TM Forum	Pierre Gauthier
Cisco	Tim McHale Joe Ruffles
Ericsson	Johannes Minnaar
Orange	Jean-Luc Tymen

8.4. Acknowledgments

This document was prepared by the members of the TM Forum Open Digital Economy team:

- Pierre Gauthier, TM Forum, **Editor**
- Johannes Minnaar, Ericsson**, Author**
- Jean-Luc Tymen, Orange, **Author**

Additional input was provided by the following people:

- Dah-Sheng Lee, ChungHwa Telecom
- Tim McHale, Cisco
- Joe Ruffles, Cisco
- Elaine Haher, Ericsson
- Sanjay Saxena, Huawei
- Wenjie Zhu, Huawei
- Nakamura Takayuki, NTT Docomo
- Tayeb Ben Meriem, Orange
- Laurence Guillon, Orange
- Veronique Mauneau, Orange
- Emmanuel Micaux, Orange
- Hervé Poupon, Orange