

[< Previous](#)[▲ Home](#)[Next >](#)

IT4IT Reference Architecture, Version 2.0
Technical Standard
Copyright © 2015 The Open Group



7 Request to Fulfill (R2F) Value Stream

This chapter provides an overview of the Request to Fulfill (R2F) Value Stream – one of four IT Value Streams that comprise the IT Value Chain. It describes the business context, objectives, and details behind the R2F Value Stream.

7.1 Objectives

The R2F Value Stream represents a modern, consumption-driven engagement model and goes beyond the traditional IT service request management. It is a framework for connecting the various consumers (business users, IT practitioners, or end customers) with goods and services that they need to drive productivity and innovation. It fosters service consumption and fulfillment, knowledge sharing, self-service support, and collaboration between communities of interest to improve the overall engagement experience with IT.

Many organizations use multiple IT requests and/or Service Catalogs to address the needs of their consumers. The R2F Value Stream brings these different catalogs and consumer personas into a single consumption experience thereby eliminating the complexity and confusion consumers experience today. The goal of the R2F Value Stream is to provide a blueprint for creating a streamlined consumption experience that consistently engages consumers and eliminates the need for them to avoid working with their IT organization.

The key to success with R2F is dependent upon two primary factors:

- The ability to package deliverables as offers that consumers recognize and value, abstracting away confusing technology choices and complex fulfillment processes. This is accomplished by leveraging the Service Model to create Service Catalog Entries that can be instantiated and consumable offers that can be requested/ordered.
- The ability to present and manage an inviting consumption experience that exposes a variety of opportunities to acquire services, goods, knowledge, and/or support. This requires organizations to go beyond the traditional IT request catalog solutions.

R2F data objects are realized through these primary functional components:

- The **Engagement Experience Portal** functional component represents the modern IT engagement/consumption experience which exposes a variety of opportunities to acquire services, goods, knowledge, and/or support.
- The **Offer Consumption** and **Offer Management** functional components present offers to consumers and enables the shopping experience. Offers are based on Service Catalog Entries developed in the Catalog Composition functional component; they are created from the consumer point of view and can be tailored to different personas, roles, or functions using profiling.
- The **Catalog Composition** functional component enables the aggregation of catalogs from multiple suppliers into a single Offer Catalog and the composition of Service Catalog Entries. Service Catalog

Entries are created from the provider point of view and may have some level of fulfillment details exposed.

- The **Request Rationalization** functional component rationalizes the order/request into individual Fulfillment Requests and authorizes Subscriptions.
- The **Fulfillment Execution** functional component routes the individual Fulfillment Requests to the appropriate fulfillment engines.
- The **Usage** and **Chargeback/Showback** functional components track the service usage and control the chargeback contract. Actual chargeback and invoicing is handled by financial systems within most companies. Further, service usage data can be used to drive continuous improvement into service offerings. Using data to provide insight into how services are being used helps shape demand for improvements and new services. This helps minimize the dependency on intrusive and time-consuming “requirements gathering sessions” with consumers.

The common limitations for current R2F practices include:

- A service consumption experience that exposes technology resources and/or IT capabilities rather than valued services.
- Multiple catalogs required for consumers to navigate in order to find and request available services.
- Too many customer service requests requiring creation of fulfillment incidents, projects, and/or human intervention resulting in delays and an unfavorable experience overall.
- Lack of service Subscription, Usage, and chargeback traceability.

7.2 Business Value Proposition

The R2F Value Stream places emphasis on time-to-value, repeatability, and consistency for consumers looking to request and obtain services from IT. It optimizes both service consumption and fulfillment experiences by delineating between the creation of offers and catalog aggregation and Service Catalog Entry composition.

Today IT organizations struggle to increase the ratio of self-sourced services over workflow-based fulfillment requiring direct human intervention. Many fulfillments today require too much intervention that consumes valuable IT resources. By increasing self-sourcing, companies will see improved business velocity and reduction in friction. They will also be able to reduce “shadow-sourcing” within the lines of business because of a more responsive consumption experience. Today’s IT is focused on delivery of technical capabilities – tomorrow’s IT must be positioned to focus on facilitating consumption of multi-sourced services.

The R2F Value Stream emphasizes the importance of deploying standard changes (a form of request management) rather than normal changes for internal fulfillments. Normal, risk-assessed and individually approved changes are one of the most time-consuming and resource-intensive activities in an enterprise-wide IT organization. Service fulfillments should be based on standard Service Models where request criteria and approvals are designed into the model and fully automated fulfillment processes orchestrate all necessary provisioning and Change Management.

The R2F Value Stream plays an important role in helping IT organizations advance toward a service broker model. Such a model is dependent on the organization's ability to leverage both internal and external sourcing options for satisfying consumer demand. Enterprise IT organizations have been using external suppliers for goods and services for many years. However, the multi-sourcing environment will become more complex as companies expand their use of cloud-based offerings. The R2F Value Stream enables the aggregation of catalogs and catalog entries from multiple providers into a single consumption experience. Therefore, while there is complexity on the delivery side in managing the various catalogs and catalog entries, it is not exposed to the consumer and the ordering experience is seamless and inviting.

The key value propositions for adopting the R2F Value Stream are:

- Provides a blueprint for increasing business innovation velocity by facilitating a service consumption experience that allows consumers to easily find and subscribe to goods and services through a self-service engagement model.
- Provides a functional framework that delineates between a single Offer Catalog and multiple Catalog Compositions to reduce complexity in the IT shopping experience.
- Provides an architectural foundation for moving from traditional IT request management to service brokerage that increases both business and IT effectiveness.
- Increased fulfillment efficiency and consistency through standard change deployment and automation.
- Provides holistic visibility and traceability across service Subscription, Usage, and chargeback to improve IT Financial Management.
- Enables increased cost optimization; for example, by canceling expired Subscriptions and reclaiming resources, Subscriptions, and/or licenses that are unused.

7.3 Key Performance Indicators

The R2F Value Stream critical success factors and Key Performance Indicators (KPIs) are as follows:

Critical Success Factors	Key Performance Indicators (KPIs)
Ability to Meet Customer Expectations	New or modified Subscriptions per time period % and number of Subscription requests complying or breaching SLA or OLA agreements Number of Subscription requests accepted and rejected by the requestor for the first time right delivery/fulfillment Variation in the average time to fulfill Subscription requests for the predictability of delivery Number of Incidents related to request fulfillment Arrival and departure rate of service requests

Critical Success Factors	Key Performance Indicators (KPIs)
Reduce Costs	<p>Costs (burned resources) per service and per fulfillment step</p> <p>Breakdown of self-source fulfillments <i>versus</i> one-off fulfillments</p> <p>% and number of fulfillments requiring human intervention to be completed</p> <p>Number of service request queues being managed</p>
External Service Provider Compliance	<p>Number of purchase orders per time period</p> <p>% and number of orders delivered and accepted complying with underpinning contract agreements</p> <p>% and number of delivered orders breaching underpinning contract agreements</p> <p>Number of Incidents related to the purchase order fulfillment</p> <p>Number of purchase orders unfulfilled at the end of a given period</p> <p>Number of orders delivered and accepted by the requestor per time period</p> <p>Number of purchase orders rejected via no delivery or cancelled purchase orders</p>
Increase Speed/Agility/Flexibility (Operational Performance)	<p>Completed service requests</p> <p>Service request work-in-progress</p> <p>Number of interactions with consumers per service during delivery</p> <p>% of work-in-progress within SLA</p> <p>% of completed work within SLA</p>

7.4 Value Stream Definition

The Request to Fulfill (R2F) Value Stream contains primary and secondary functional components. Primary functional components are core to the value stream and are essential for managing the service at this stage of its lifecycle. Secondary functional components are not dedicated to R2F but provide relevant data objects to primary functional components. R2F includes the following primary functional components that provide the technology necessary to support IT portfolio activities:

- Offer Consumption
- Offer Management
- Catalog Composition

- Request Rationalization
- Fulfillment Execution
- Usage
- Chargeback/Showback
- Knowledge & Collaboration

A functional component utilizes an input data object to conduct key activities and may also provide output data objects to another functional component that requires that data.

The R2F Value Stream primarily focuses on system of record integrations between two functional components like other value streams. However, the R2F Value Stream also includes system of engagement integrations enabling a common user experience or graphical user interface mash-up. The R2F Value Stream functional components with system of engagement integrations are the Engagement Experience Portal and Offer Consumption functional components. The Offer Consumption functional component integrations include system of engagement integrations with the following functional components: Offer Management, Request Rationalization, and Chargeback/Showback.

The Engagement Experience Portal functional component is a secondary functional component since its scope can expand beyond the IT4IT Reference Architecture and into the business domain. It exposes and facilitates a unified engagement between consumers and the IT functions. The main objectives of the Engagement Experience Portal are to:

- Drive the consumption through the Offer Catalog
- Enable collaboration between communities of interest
- Obtain support through a self-service interface
- Access knowledge that enables them to be better informed about services offered by IT

There may be other engagement experiences presented through the portal but this is the core set called out explicitly in the R2F Value Stream.

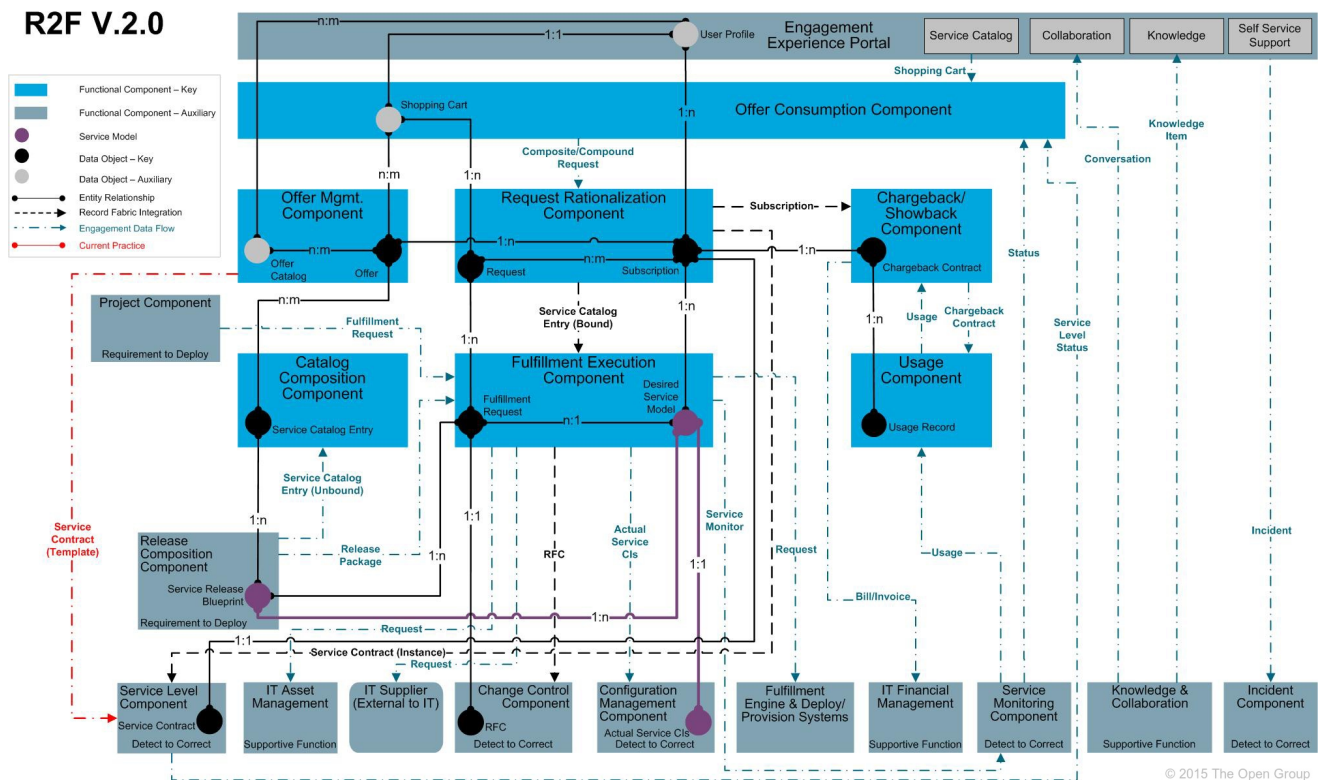


Figure 55: Request to Fulfill Level 2 Value Stream Diagram

7.4.1 Engagement Experience Portal (Secondary Functional Component)

Purpose

- The Engagement Experience Portal functional component is based on a system of engagement integration design pattern where consumers access different functional components through a common user experience.
- Consumers manage certain aspects of their profile; for example, localization, preferred method of communication, user interface settings, and so on.
- Facilitate service consumption by connecting any potential consumer with the right information, goods, services, or capability at the right time through a single experience, taking into account the consumer profile.
 - Provide an intuitive experience that draws consumers in instead of being viewed as an inhibitor to productivity that is forced upon them.
 - Provide a self-configurable experience (such as mash-ups) so that the different types of consumers can tune the experience to best suit their needs.
 - Provide an interface supported across multiple devices such as smartphones, tablets, etc.
 - Provide plug-and-play connectivity for components that need to be exposed through the portal.
 - Components for connectivity include but are not limited to catalog-driven service consumption.
- Self-service support:

- Community and collaboration
- Knowledge and content associated with services and capabilities
- Information about consumed services
- Information from the Service Monitoring and the Service Level functional components (D2C Value Stream) such as the (current) service status

Key Data Objects

- **User Profile** (data object): Personal data associated with a specific user and the explicit digital representation of a person's identity. A User Profile consists of different attributes managed by a user or consumed by other authoritative sources. User Profile attributes must be secure, protected, and access restricted based on roles (e.g., HR Manager) or system.

Key Attributes

The User Profile data object shall have the following key data attributes:

- **UserID**: Unique Identifier of the user.
- **UserName**: Full name of the user.
- **Role**: Used to grant access to functionality and information.

Key Data Object Relationships

The User Profile data object shall maintain the following relationships:

- **User Profile to Offer Catalog** (n:m): Present a personalized list of offers from the catalog depending on the consumer profile.
- **User Profile to Shopping Cart** (1:1): The catalog items which are ordered need to link to the consumer that will receive the items. If an approval step is required, the User Profile helps to identify the authorized person by looking up information from the organizational hierarchy.
- **User Profile to Subscription** (1:n): A Subscription is created and linked to the user for every service that has been ordered and fulfilled where a Subscription is required.

Main Functions

The Engagement Experience Portal functional component:

- Shall be available to all users that desire to consume IT services.
- Shall expose various IT functions and capabilities in a single place, unifying the experience. These functions and capabilities may be exposed in a form similar to smartphone apps.
- May allow consumers to manage their User Profile (to varying degrees as some attributes may be provider-controlled).

Model

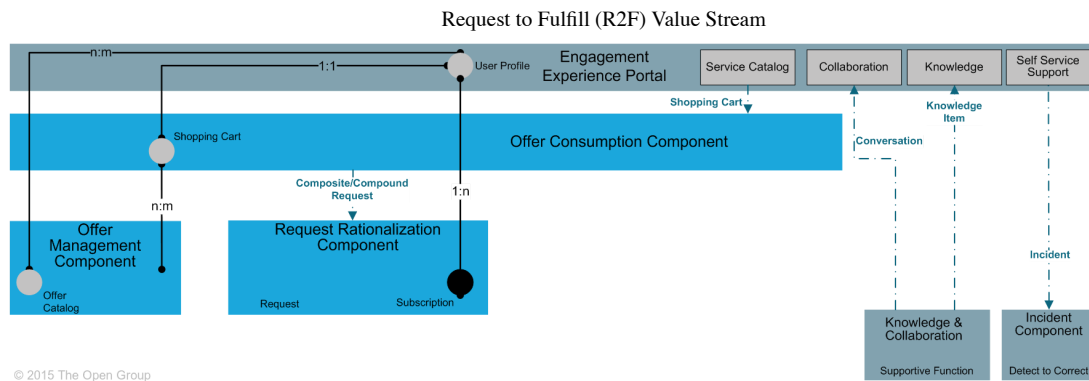


Figure 56: Engagement Experience Portal Level 2 Model

7.4.1.1 Service Catalog Functional Sub-Component

The Service Catalog functional sub-component enables consumers to engage with and consume services through the Offer Consumption functional component. This includes but is not limited to ordering new services, modifying Subscriptions, viewing the status of existing services or requests, and costs associated with services. For a detailed description of the functionality of the Service Catalog, refer to [Section 7.4.2](#).

7.4.1.2 Collaboration Functional Sub-Component

The Collaboration functional sub-component provides the user front end for an enterprise collaboration experience, such as a chat capability.

7.4.1.3 Knowledge Functional Sub-Component

The Knowledge functional sub-component provides the interface for users to search and read Knowledge data objects of all types and sources. Knowledge data objects may include but are not limited to IT or supplier-created technical briefs, training videos, and user-created content.

7.4.1.4 Self-Service Support Functional Sub-Component

The Self-Service Support functional sub-component:

- Provides service consumers with a way to address more of their IT-related issues, as well as receive information regarding their existing records without necessarily engaging IT providers
- Reduces the load on the IT support organization by enabling and promoting self-help and self-healing behavior through the use of communities, knowledge sharing, content, etc.

The Self-Service Support functional sub-component contains the following data object:

- **Incident:** Create an Incident in the Incident functional component. For a detailed description, refer to [Section 8.4.3](#).

The Self-Service Support functional sub-component:

- Shall enable users to create new support tickets for issues and/or questions that they were not able to resolve, or their questions that remain unanswered.
- Shall enable users to view and update their existing support tickets.
- Can route users to access the Knowledge data objects before a new support ticket is created.

- Can incorporate the Collaboration functional sub-component to enable communication with peers and IT staff in order to gain knowledge or resolve their IT-related issues.

7.4.2 Offer Consumption Functional Component

Purpose

- Present service offers to various service consumers; facilitate consumption/management of and payment for IT services rendered.
- Utilize the User Profile to present a personalized experience which includes consumer-specific information such as personal preferences, location, or job function.
- Present consumable offers derived from Service Catalog Entries with associated descriptions, pictures, prices, and purchasing options to prospective consumers.
- Enable consumers to manage their Subscriptions by:
 - Viewing Subscription/service status, costs, usage, etc.
 - Adding new Subscriptions, ordering new services, or service instances
 - Modifying Subscription parameters (also known as upgrading/downgrading)
 - Cancelling/ending service Subscriptions

Key Data Objects

- **Shopping Cart** (data object): Contains the IT services that the user wants to order; the object only exists during the actual shopping session. Upon submission, a request is generated that is comprised of the content contained in the Shopping Cart.

Note: Shopping Cart is an auxiliary data object (not key) but is being described here to help with the reader's understanding of the R2F Value Stream.

Key Attributes

The Shopping Cart data object shall have the following key data attributes:

- **ShoppingCartID**: Unique identifier for the Shopping Cart.
- **UserID**: Unique identifier for the user (from User Profile).
- **ApproverID**: Based on the user, the identifier of the approver (e.g., user ID of the manager).
- **Status**: Controls the status of the approval steps/workflow if needed.

Multiple services can be ordered for every item ordered in one Shopping Cart, therefore these attributes apply per item in the Shopping Cart:

- **LineItem**: Multiple items can be ordered in one go.
- **OfferID**: Identifier of the Offer (from Offer).
- **ReqValue**: Based on the Offer the user might need to provide values/options for a successful fulfillment (from Offer).

Key Data Object Relationships

The Shopping Cart data object shall maintain the following relationships:

- **Shopping Cart to User Profile (1:1):** The contents of the Shopping Cart relate to a specific user, who is actually ordering the services.
- **Shopping Cart to Offer (n:m):** The Shopping Cart only contains those items available to the end-user from the existing Offers.
- **Shopping Cart to Request (1:n):** Represents the relationship between the Shopping Cart and the Requests necessary to fulfill the services ordered in the shopping experience.

Main Functions

The Offer Consumption functional component:

- Can provide information on the existing Subscription to enable the user to change/cancel existing Subscriptions.
- Shall provide all necessary information to guarantee the fulfillment.
- Can provide functionality to order multiple offers in one transaction.
- Can enable consumers to order services on behalf of other consumers.
- Can provide visibility to information about the user's specific service consumption including pricing, usage, etc.

If the Service Level functional component exists, the Offer Consumption functional component:

- Shall expose information on the Service Level status for the services the user subscribed to.

Model

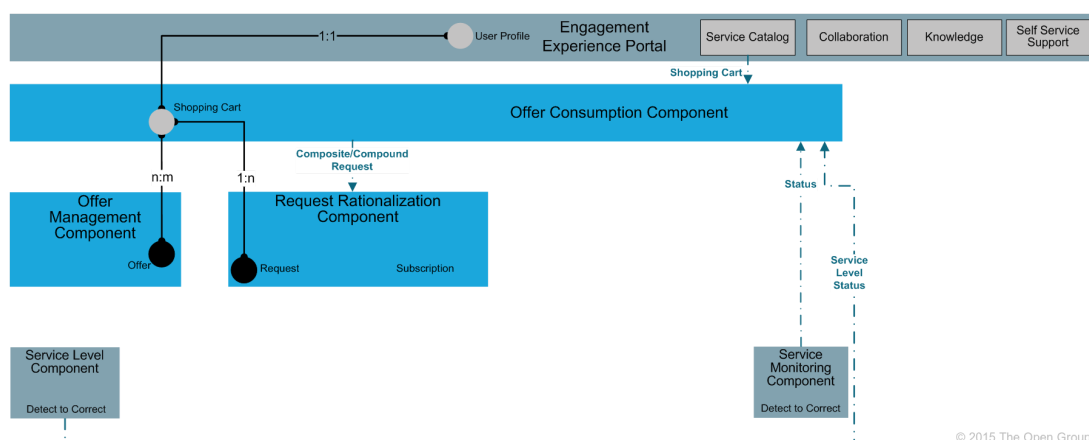


Figure 57: Offer Consumption Functional Component Level 2 Model

7.4.3 Offer Management Functional Component

Purpose

- Aggregate (mash-up) all Catalog Composition items and external supplier catalogs into consumable Offers that users can order through the Offer Consumption functional component.
- Build and publish the various offerings into Offer Catalogs for various populations to consume, determine prices, and valid options that consumers can select.
- Enable Offers to be grouped into an Offer Catalog to expose them as a collection of consumable items for a given group of consumers.
- Fulfill each Offer through numerous underlying Catalog Compositions as determined by this functional component.

Key Data Objects

- **Offer** (data object): Defines how a Service Catalog Entry will be instantiated and under what terms and conditions – price, deployment, approval, workflow, service level (contract), etc.
- **Offer Catalog** (data object): A set or collection of Offers that are grouped together as something that can be consumed by certain consumers or consumer groups.

Key Attributes

The Offer data object shall have the following key data attributes:

- **OfferID**: Unique identifier for every Offer in the catalog.
- **CatalogID**: Identify in which catalog this Offer is available (from Offer Catalog).
- **OfferName**: Description of the Offer for consumers to identify/search Offers.
- **StartDate**: Date/time on which the Offer may be consumed.
- **ExpiryDate**: Date/time on which the Offer is no longer available.
- **Status**: Indicates if the Offer is ready for consumption (e.g., draft, published, retired, etc.).
- **Price**: If applicable, the pricing information on the service, including the type of Subscription.
- **ReqValue**: Mandatory options or variables linked to the service which need to be provided by the consumer to prevent issues during the fulfillment. (Some of) these options or variables might not be selectable for customers, but are pre-filled by the Offer itself upon creation of the Offer.

The Offer Catalog data object shall have the following key data attributes:

- **CatalogID**: Unique identifier for the Offer Catalog.
- **CatalogName**: Offer Catalog name used for consumers.
- **ServiceID**: Identifier for the service (from Service Catalog Entry).
- **Roles**: Authorization role required for access (from User Profile).

Key Data Object Relationships

The Offer data object shall maintain the following relationships:

- **Offer to Service Catalog Entry (n:m):** Ensures all required information is captured for the fulfillment (deployment/delivery) of the service.

The Offer Catalog data object shall maintain the following relationships:

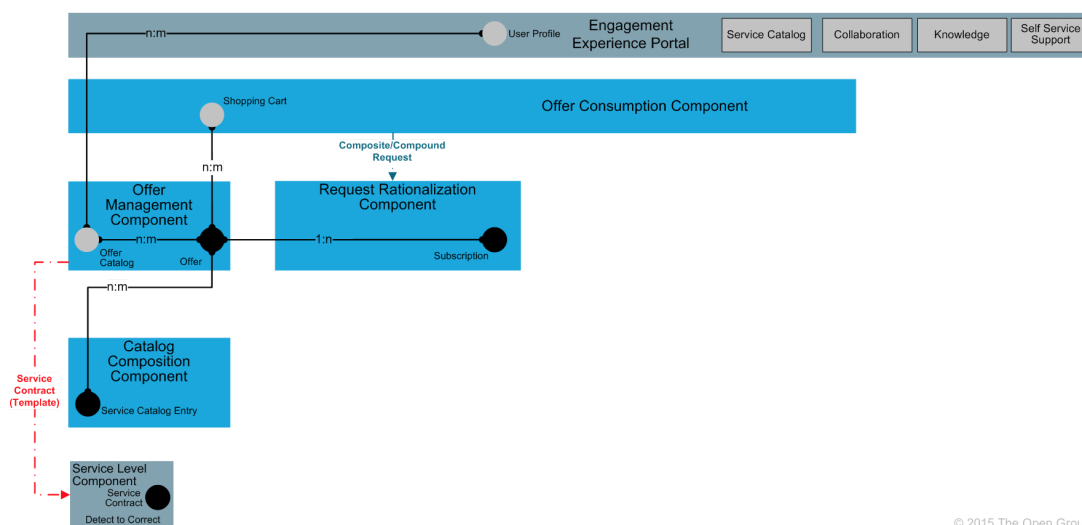
- **Offer Catalog to Offer (n:m):** Represents the collection of Offers that comprise each Offer Catalog.
- **Offer Catalog to User Profile (n:m):** Represents which users can access/consume each Offer Catalog.

Main Functions

The Offer Management functional component:

- Shall contain all of the Offers available to consumers and provide this information to the Offer Consumption functional component.
- Can group services from multiple providers (internal and external) into a single Offer (also known as a bundle of services).
- May create the Service Contract template and provide information to the Service Level functional component.

Model



© 2015 The Open Group

Figure 58: Offer Management Functional Component Level 2 Model

7.4.4 Catalog Composition Functional Component

Purpose

- Create and publish the Service Catalog Entries, including all of their dependencies, at the level at which these can be presented as Offers in the Offer Management functional component.
- Service Catalog Entries are created from the Service Release Blueprint in the Release Composition functional component (R2D Value Stream). Services, as well as their dependencies and details, are accurately defined, including supplying the necessary information for the service to be instantiated.
- Service Catalog Entries are created and updated to prepare them for consumption, including configurable options (e.g., pricing, subscription terms, bundles, service level, support conditions, etc.).

Key Data Objects

- **Service Catalog Entry** (data object): A Logical Service Blueprint within the R2F Value Stream managed by the Offer Management functional component. It is an authoritative source for the consolidated set of IT services that can be presented as Offers. An unbound part of the superset of IT service definitions is available for promotion to the status of Offer. A Service Catalog Entry would need to be bound to aspects relating to the delivery of the IT service such as price, deployment, approval, workflow, and service level in order to be promoted to Offer.

Key Attributes

The Service Catalog Entry data object shall have the following key data attributes:

- **ServiceID**: Unique identifier for every service in the catalog.
- **ServiceName**: Name of the service used for catalog and offer creators.
- **ReqValue**: Provide a set of values/variables that are needed upon fulfillment.

Key Data Object Relationships

The Service Catalog Entry data object shall maintain the following relationships:

- **Service Catalog Entry to Service Release Blueprint** (1:n): Ensure all catalog entries relate to the specific service definitions used for fulfillment.
- **Service Catalog Entry to Offer** (n:m): Ensure all information needed is captured during the order phase.

Main Functions

The Catalog Composition functional component:

- Shall manage inter-dependencies within the services.

Model

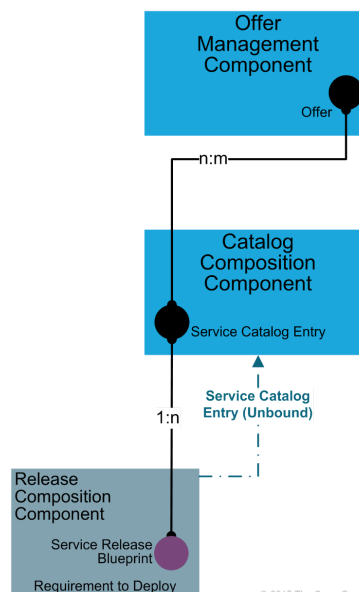


Figure 59: Catalog Composition Functional Component Level 2 Model

7.4.5 Request Rationalization Functional Component

Purpose

- Rationalize, break down, and route “clean order” requests (ready for fulfillment) to appropriate Fulfillment Execution engines or providers in order to deliver services to consumers. May involve breaking down a single order/request into multiple Fulfillment Requests.
- Ensure appropriate fulfillment-related Subscription information is kept up-to-date, such as approval/rejections, modifications, cancellations, and so on.
- Enable the recording of patterns of service consumption that can be used to shape demand for new and/or improved services.
- Break the request down into the IT services and provide these to the Fulfillment Execution functional component and create the Subscriptions for these services upon their successful fulfillment.
- Fulfillment status is tracked and completion notifications from fulfillment channel(s) are received. Consumers will be able to receive status updates at the Subscription level, not at the level of the underlying requests needed to fulfill the Subscription.

Key Data Objects

- **Request** (data object): Contains all Offers from the Shopping Cart which have been consumed and need to be fulfilled.
- **Subscription** (data object): Managed by the Request Rationalization functional component. This data object represents the rights to access a service that has been provided to a consumer.

Key Attributes

The Request data object shall have the following key data attributes:

- **RequestID**: Unique identifier for the Request.
- **UserID**: User identifier (from Shopping Cart).
- **Status**: Controls the status of the fulfillment.
- **RequestDate**: Date/time the Request was received.
- **MaxFulFillDate**: Maximum date/time on which the Request needs to be fulfilled.
- **ActFulFillDate**: Date/time on which the Request is fulfilled.
- **OfferID**: Identifier of the service (from Offer).
- **SubscriptionID**: Link to the Subscription for this service (from Subscription).
- **ServiceID**: Based on the Offer, the service(s) are identified (from Offer).
- **ReqValue**: Based on the Offer, the user might need to provide values/options for a successful fulfillment (from Offer).

The Subscription data object shall have the following key data attributes:

- **SubscriptionID**: Unique identifier for the Request.
- **UserID**: Unique identifier of the user (from Shopping Cart).

Key Data Object Relationships

The Request data object shall maintain the following relationships:

- **Request to Shopping Cart** (1:n): Enables the traceability of Requests to the originating order (in the form of the Shopping Cart) which is used to report the current status of the order.
- **Request to Subscription** (n:m): Enables traceability between the Request and the resulting Subscription. This helps to better understand how the current Subscription state was realized.
- **Request to Fulfillment Request** (1:n): Used for tracking fulfillment as well as to navigate between dependent Fulfillment Requests.

The Subscription data object shall maintain the following relationships:

- **Subscription to User Profile** (n:1): Enables the consumer to manage (view/modify) all of their Subscriptions. Also helps provide important information related to the specific consumer Subscription.
- **Subscription to Offer** (1:n): Provides traceability between the Subscription and the Service Contract (via the Offer).
- **Subscription to Chargeback Contract** (1:n): Facilitates the various chargeback/showback calculations that are dependent on Subscription details such as its contract duration and service status.
- **Subscription to Desired Service Model** (1:n): Enables traceability between the consumer, their Subscription, and the realized Service Model.

Main Functions

The Request Rationalization functional component:

- Shall provide information to the consumer on the fulfillment status.
- Shall provide Subscription information for the creation of the associated Chargeback Contract.
- Shall provide information on Request delivery times for SLA measurements.
- Shall break down the composite request (described by the Shopping Cart and consumer-selected values) into the individual Requests that need to be fulfilled.
- Shall send the bound Service Catalog Entry to the Fulfillment Execution functional component in order for it to create the Fulfillment Requests needed to satisfy the order.

Model

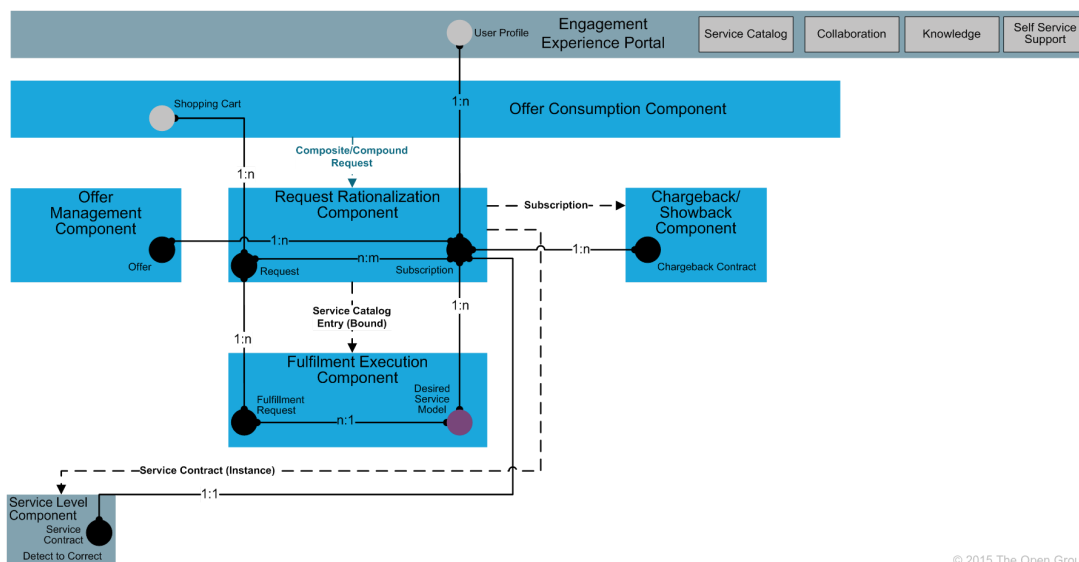


Figure 60: Request Rationalization Functional Component Level 2 Model

7.4.6 Fulfillment Execution Functional Component

Purpose

The Fulfillment Execution functional component orchestrates the delivery of the various requests amongst (one or more) fulfillment engines in order to deliver the IT service. Fulfillers may be systems that perform actions directly, or engage other systems in order to perform actions. They may also include external providers. In order to be able to engage the fulfillers, the Fulfillment Execution functional component needs to:

- Manage a registry of the available fulfillers. This registry captures:
 - What each fulfiller does (capabilities)
 - How to engage each fulfiller (where they are located and how to invoke them)
- Take the bound Service Catalog Entry and generate both the relevant Fulfillment Requests in order to realize/fulfill the originating consumer request and the Desired Service Model data object which represents the Service Model in its pre-configured or consumer configured state.
- Update the IT asset inventory as they are ordered.
- Request standard changes and update the Configuration Management functional component (if needed) on delivery of components.
- Maintain visibility into supplier capacity levels and raise alerts if capacity appears to be insufficient for immediate demand.

The Fulfillment Execution functional component can be used via two paradigms:

- **Consumer-driven** – In this paradigm a consumer request results in a bound Service Catalog Entry which is broken down into the necessary Fulfillment Requests needed to fulfill the originating request.
- **Direct access** (without a Service Catalog Entry) – In cases in which there aren't sufficient catalog entries to describe the fulfillment and no entries are planned to be created, the Fulfillment Execution functional component is directly engaged by the Release Composition functional component (R2D)

Value Stream). The Release Composition functional component needs to provide enough information in order for the Fulfillment Execution functional component to create the Fulfillment Request(s) necessary to perform the actions needed.

Key Data Objects

- **Fulfillment Request** (data object): Describes all fulfillment aspects of an IT service, which includes items such as provisioning, deploying, modifying, actions (i.e., start, stop, etc.), decommissioning, and so on.
- **Desired Service Model** (data object): This is an instantiation of the unbound Service Catalog Entry, which is the binding of the relevant parameters that determine how a service will be deployed/fulfilled. This results in a single realized deployment for the service. The parameters are set by the user's selections made in the Offer Consumption functional component, as well as the determinations made in the design of the service that are interpreted by the Fulfillment Execution functional component. This data object:
 - Validates that the realized service matches the selections in the original request and notes exceptions or modifications where applicable
 - Will comply with the definitions of the Service Catalog Entry

Note: The realized service (Actual Service CI) must also comply with the policies governing the IT environment.

Key Attributes

The Fulfillment Request data object shall have the following key data attributes:

- **FulfillmentID**: Unique identifier for the Request.
- **RequestID**: Refers to the originating Request (from Request).
- **DesiredServiceID**: Links to the service to be instantiated (from Desired Service Model).
- **RFCID**: If applicable, an RFC will be created (from RFC).
- **Status**: Status of the deployment/fulfillment workflow.

The Desired Service Model data object shall have the following key data attributes:

- **DesiredServiceID**: Unique identifier for the bound IT service to be delivered.
- **SubscriptionID**: Tracability to the Subscription (user of the IT service) (from Subscription).
- **ServiceReleaseBlueprintID**: Getting necessary information on the IT service (from Service Release Blueprint).
- **ActualServiceCI_ID**: Link to the Configuration Management functional component (from Actual Service CI).

Key Data Object Relationships

The Fulfillment Request data object shall maintain the following relationships:

- **Fulfillment Request to Request** (n:1): Inform on Fulfillment Request status.
- **Fulfillment Request to Service Release Blueprint** (n:1): The Service Release Blueprint supplies the Fulfillment Request with information needed to instantiate the service.
- **Fulfillment Request to Desired Service Model** (n:1): Acquire relevant information.
- **Fulfillment Request to RFC** (1:1): If applicable, the RFC created can be linked to the originating Request.

The Desired Service Model data object shall maintain the following relationships:

- **Desired Service Model to Subscription** (n:1): Create the traceability from service to Subscription.
- **Desired Service Model to Service Release Blueprint** (n:1): Acquire all necessary service information for fulfillment.
- **Desired Service Model to Actual Service CI** (1:1): Create traceability and enable verification of correct deployment/fulfillment.

Main Functions

The Fulfillment Execution functional component:

- Shall select the appropriate fulfillment mechanism.
- Shall coordinate if multiple fulfillment mechanisms are needed and manage the dependencies required to fulfill the IT service request.
- Shall provide the Subscription status to the Request Rationalization functional component.
- Shall create the Actual Service CIs within the Configuration Management functional component.
- May create an RFC associated with the service instantiation that is created within the Change Control functional component (D2C Value Stream). The RFC type, standard or normal, is determined within the Change Control functional component.
- Can create a new service monitor or modify an existing one for the service provided in the Request as part of fulfillment.
- Can create/route a Request to an external service provider to fulfill part or all of the service.
- Can request IT assets necessary for fulfillment (such as licenses). This also enables the tracking of assets being requested or procured and links them with the services that require them.
- Can trigger deployment engines to enable fulfillment of the service.

Model

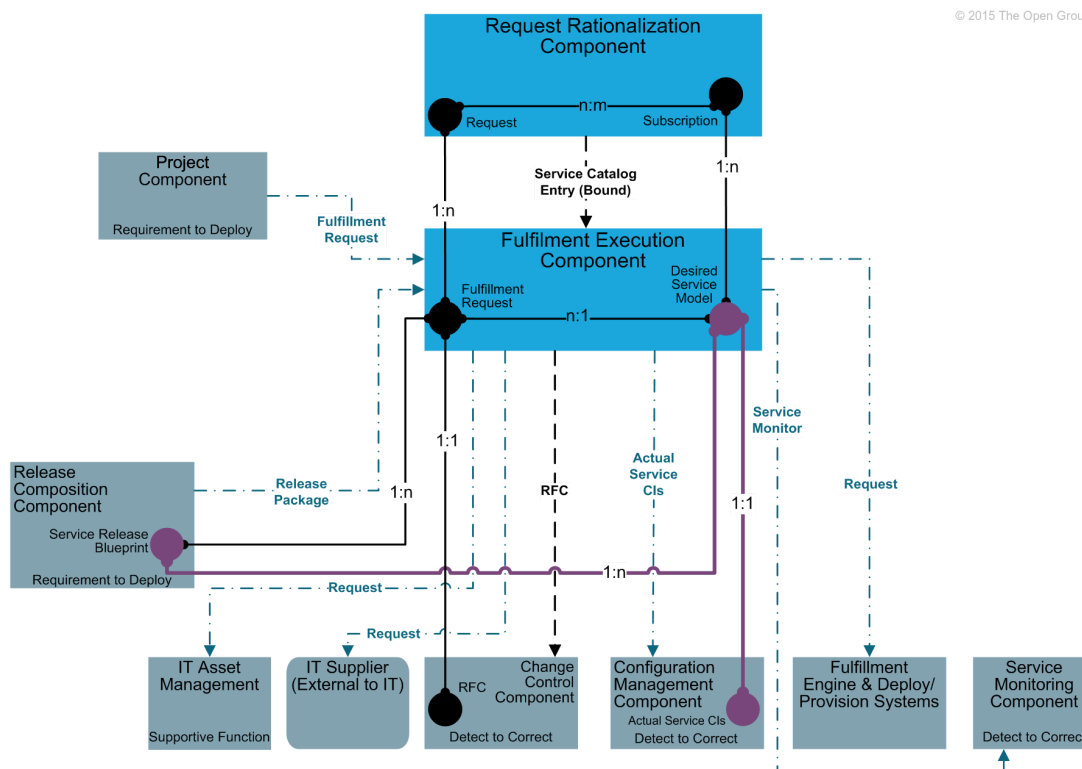


Figure 61: Fulfillment Execution Functional Component Level 2 Model

7.4.7 Usage Functional Component

Purpose

- Track actual usage of subscribed IT services by gathering IT service usage metrics, activity, and history for both internal and external sourced IT services.
- Process and break down usage information for each Subscription, its consumers (singular, group), provider, etc.
- Collect internally and externally-sourced IT service usage metrics from the Service Monitoring functional component (D2C Value Stream).

Key Data Objects

- **Usage** (data object): A data object within the R2F Value Stream managed by the Usage functional component. It is the measured use of a particular service or service component. An example usage can be composed of (internal) hours, system usage (capacity, CPUs, etc.), or external supplier usage.

Key Attributes

The Usage data object shall have the following key data attributes:

- **UsageID:** Unique identifier for the service Usage.
- **ChargebackContractID:** Relate the service Usage to the Chargeback Contract.

Key Data Object Relationships

The Usage data object shall maintain the following relationships:

- **Usage Record to Chargeback Contract (n:1):** Usage records are utilized to calculate chargeback amounts in cases in which the Chargeback Contract is dependent on Usage.

Main Functions

The Usage functional component:

- Shall encrypt sensitive usage information or set appropriate access controls.
- Can generate service usage history and activity reports.
- Can provide usage information to the Chargeback Contract component enabling usage-based showback or chargeback.

Model

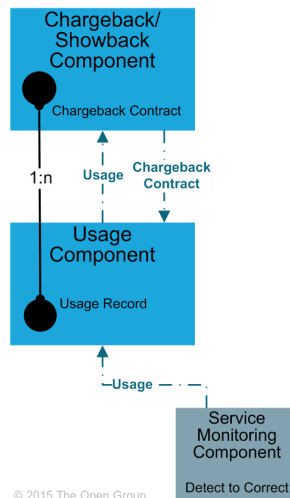


Figure 62: Usage Functional Component Level 2 Model

7.4.8 Chargeback/Showback Functional Component

Purpose

- Provide chargeback or showback for internal and external services taking into account service Subscription and Usage information.
- Break down chargeback or showback based on charge types such as CapEx/OpEx, direct/indirect, fixed/variable, and labor/non-labor, as well as by consumer or consumer groups.
- Charge for IT services rendered based on the pricing model associated with the catalog item, this may also be implemented via showback (no actual billing or cross-charging), which helps the consumer become more IT cost-aware.
- Trace chargeback or showback line items to individual cost drivers based on the cost model.
- Consolidate IT service Subscription (right to use) and actual Usage as the usage may differ from the right to use.

Key Data Objects

- **Chargeback Contract** (data object): Details the financial obligations between the service consumer and provider(s). The Chargeback Contract typically defines the unit of measure (price) for a given service and is often tightly linked to Usage.

Key Attributes

The Chargeback Contract data object shall have the following key data attributes:

- **ChargebackContractID**: Unique identifier for the Service Contract.
- **SubscriptionID**: Link to the Subscription for which this Service Contract is instantiated (from Subscription).

Key Data Object Relationships

The Chargeback Contract data object shall maintain the following relationships:

- **Chargeback Contract to Subscription** (n:1): This relationship provides the traceability between the service rendered (represented by the Subscription) and the expected charges for those services (described in the Chargeback Contract).

Main Functions

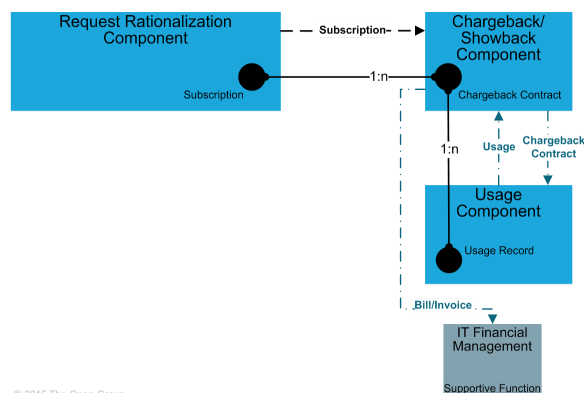
The Chargeback/Showback functional component:

- Shall provide the price of consuming/subscribing to a service.
- Can take actual usage into consideration when calculating the price of consuming a service.

If an IT Financial Management component exists, the Chargeback/Showback functional component:

- Can provide the necessary information in order for the IT Financial Management supporting function to produce invoices or bills for services rendered.

Model



© 2015 The Open Group

Figure 63: Chargeback/Showback Functional Component Level 2 Model

7.4.9 Knowledge & Collaboration Supporting Function

Purpose

Note: In the transition from Version 1.3 to 2.0 of the IT4IT Reference Architecture this functional component became a supporting component. However, it remains an important part of the R2F Value Stream in that it facilitates self-sufficiency across the consumer base. In future releases this section may be moved to later in the document but was kept here to facilitate readability.

- Provide knowledge in the form of content and Conversations that help to address the needs of IT service consumers. Knowledge includes structured IT/supplier produced articles, or unstructured Conversations from business/IT users, webinars, videos, training materials, etc. which are searchable by the IT service consumers.
- Increase the contribution to knowledge by providing all users with the ability to generate new content, either through informal Conversations, or by more formal submissions of knowledge.
- Encourage contributions by promoting a collaborative culture through various techniques such as gamification.
- Improve accessibility of knowledge in the organization by:
 - Supporting key word search capabilities
 - Providing filter capabilities based on various attributes of the knowledge, such as subject category, time range, source types (internal *versus* external), etc.
 - Supporting natural language queries to reduce the complexity of finding relevant information
 - Providing users with access to third-party knowledge and forums
 - Providing natural language processing analytics so (for example) “trending topics” can be reported from service desk interactions
- Reduce the number of requests for information/knowledge that arrive at the IT service desk.
- Provide knowledge for consumption by additional value streams in general and specifically by the D2C Value Stream.
- IT staff participate in Conversations related to IT services that they plan, develop, or operate.
- IT service consumers and IT staff consume third-party knowledge through the same experience as the formal and informal forms of knowledge the company provides.

Key Data Objects

- **Knowledge** (data object): Structured and unstructured Knowledge from the Knowledge & Collaboration functional component.
- **Conversation** (data object): User Conversations from the Knowledge & Collaboration functional component.

Key Attributes

The Knowledge data object shall have the following key data attributes:

- **KnowledgeID**: Unique identifier of the Knowledge.

- **AuthorID:** Unique identifier of the responsible author.
- **RelatedService:** If provided, linked to the Actual Service CI in Configuration Management.
- **ProblemID:** If applicable, a link to a related Problem will be provided.
- **Status:** For example, draft, revise, published, review, archived.
- **PublishDate:** Date/time on which the item is available for publication.
- **ExpiryDate:** Date/time on which the item is no longer visible.
- **Title:** Title of Knowledge data object.
- **Body:** Body text, video, or any other content that is published.

The Conversation data object shall have the following key data attributes:

- **UserID:** Unique identifier of the responsible author.
- **KnowledgeID:** Related Knowledge data object(s).
- **Body:** Body text, video, or any other content.

Key Data Object Relationships

The Knowledge data object shall maintain the following relationships:

- **Knowledge to Problem (n:m):** If a link to a Problem exists, the Knowledge data object will need changing when a Problem is (partly) resolved.

The Conversation data object shall maintain the following relationships:

- **Knowledge to Conversation (n:m):** Conversations can be linked to the Knowledge data object(s). A Knowledge manager is typically responsible for ensuring the content creates added value.

Main Functions

The Knowledge & Collaboration component:

- Shall enable SMEs to submit and/or approve Knowledge data objects (whether they are IT staff or service consumers).
- Shall provide functionality to enable the IT service consumers and IT staff to rank Knowledge data objects and Conversations, thus improving future knowledge consumption.
- Shall provide functionality to enable IT service consumers to participate in Conversations relating to the IT services they consume (collaboration).
- Can aggregate multiple (internal and external) Knowledge sources.

Model

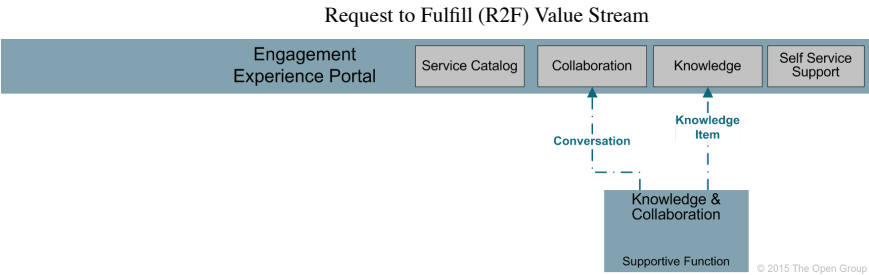


Figure 64: Knowledge & Collaboration Supporting Function Level 2 Model

Copyright © 2015 The Open Group, All Rights Reserved
IT4IT is a trademark of The Open Group.