IT4IT™

Definitions and Symbols

THE Open GROUP

Information Model

Object Type	Description	Informal Representation	Formal Representation
Key Data Object	Key data objects describe aspects of “how” services are created, delivered, and consumed; they are essential to managing the service lifecycle. Managing the end-to-end service lifecycle and associated measurement, reporting, and traceability would be virtually impossible without them. The IT4IT Reference Architecture defines 32 key data objects and most are depicted as black circles.	● Key Data Object	Data Object
Service Model	Service models are a stand-alone subclass of key data objects that describe “what” IT delivers to its consumers. They represent the attributes of a service at three levels of abstraction: Conceptual, Logical, and Realized. These data objects are referred to as Service Model Backbone data objects (or service backbone data objects in short form) and depicted using a purple colored circle in the IT4IT Reference Architecture diagrams.	🟣 Service Model	Data Object
Auxiliary Data Object	Auxiliary data objects provide context for the “why, when, where, etc.” attributes and, while they are important to the IT function, they do not play a vital role in managing the service lifecycle. The IT4IT Reference Architecture currently describes eight (8) auxiliary data objects and they are depicted using a gray colored circle.	⚪ Auxiliary Data Object	Data Object

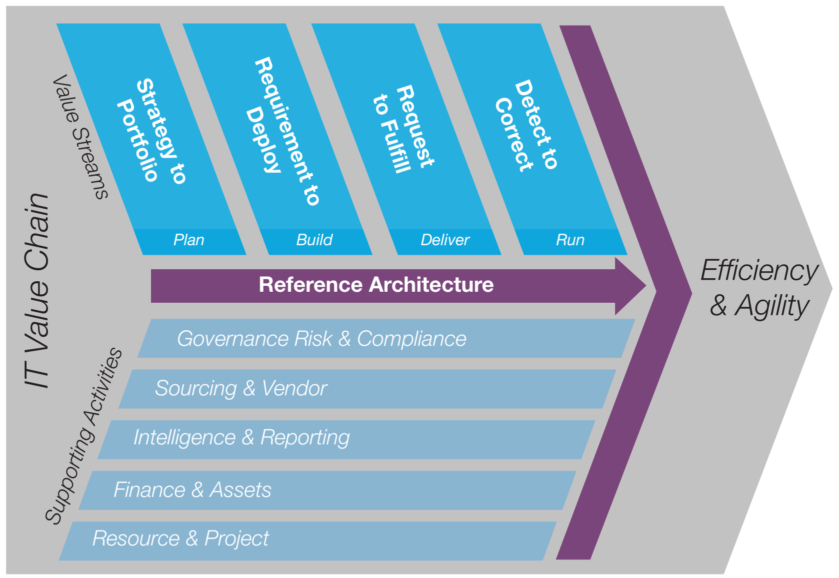
Functional Model

Object Type	Description	Informal Representation	Formal Representation
Primary Functional Component	A primary functional component is depicted using a blue colored rectangle and is core to a specific value stream. This means that the functional component plays a key role in the activities of a particular value stream. Without this functional component, the integrity of the data objects and thus the Service Model could not be maintained consistently and efficiently. Most IT4IT documentation will use language such as “a functional component is owned by or is core to a particular value stream” to represent a primary functional component.	Functional Component (Primary)	Application Component
Secondary Functional Component	Secondary functional components are depicted using a gray colored rectangle and represent some level of dependency or interaction with a value stream and its data objects. While they interact with a value stream, they are not core to it and are either primary to another value stream or supporting function or represent a capability	Functional Component (Secondary)	Application Component

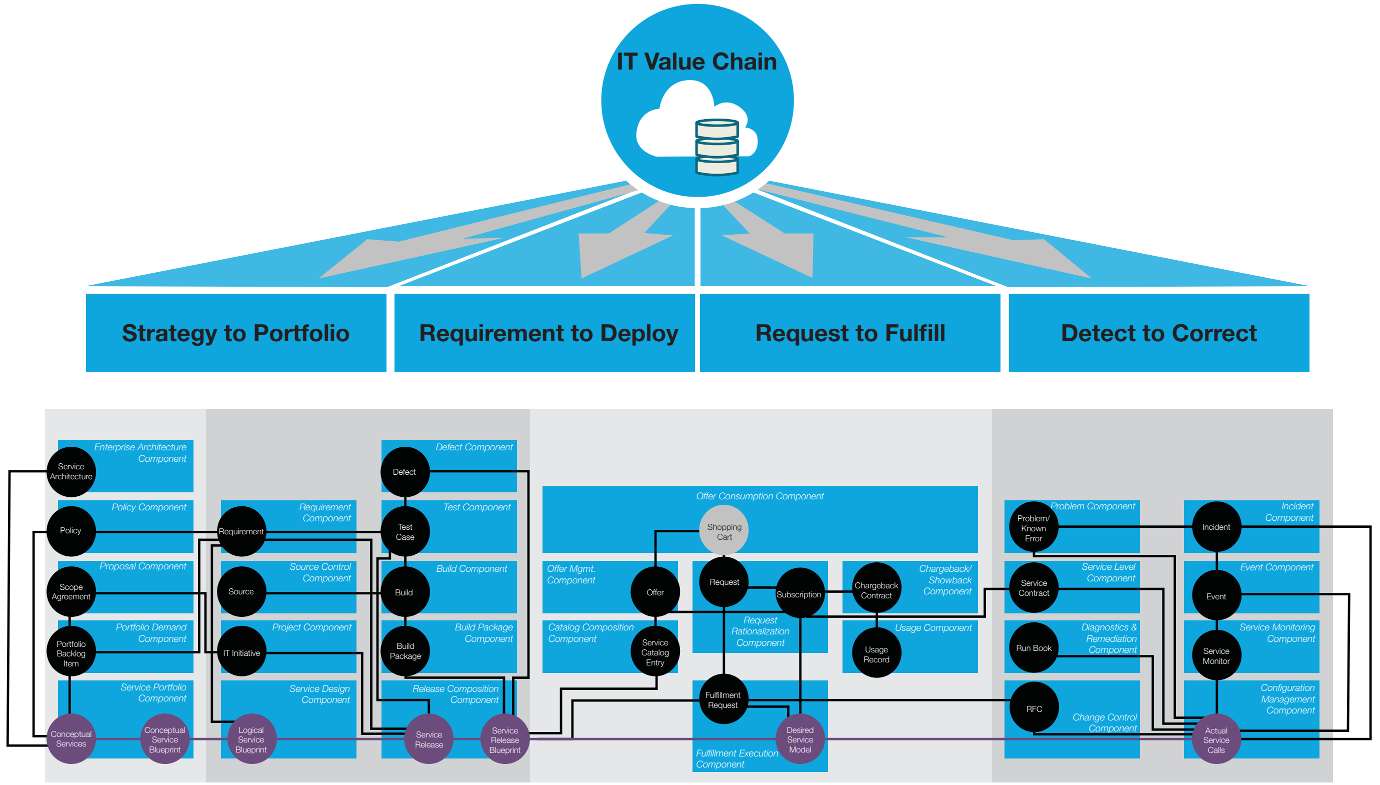
IT4IT™ Levels 1 and 2

THE Open GROUP

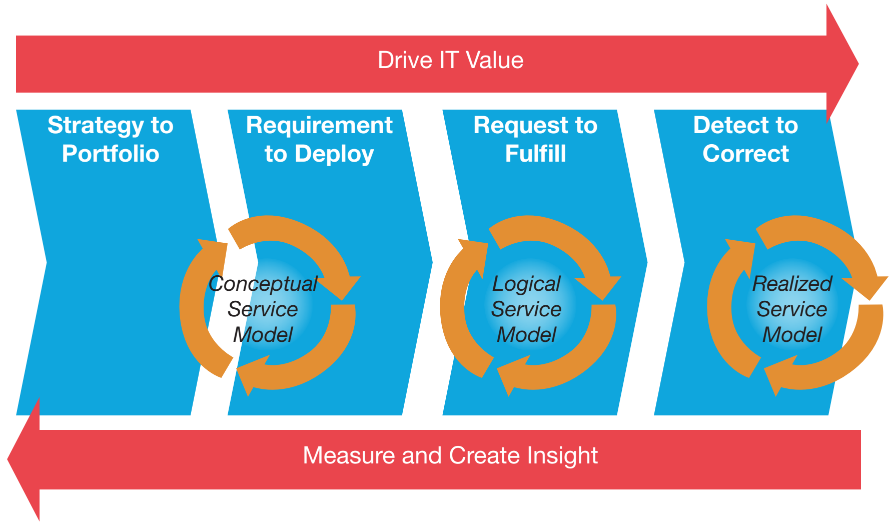
Leveraging Business Value Chain Success

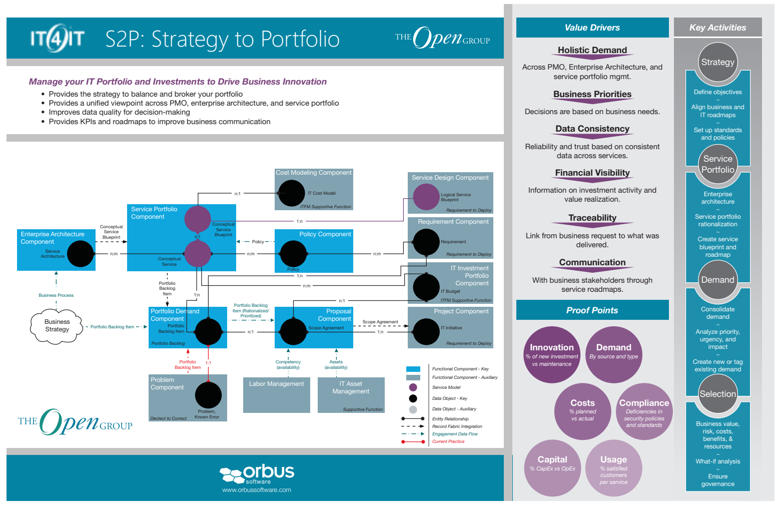


Value Stream Overview



Service Model Lifestyle





IT4IT S2P: Strategy to Portfolio

THE Open GROUP

Manage your IT Portfolio and Investments to Drive Business Innovation

- Provides the strategy to balance and broker your portfolio
- Provides a unified viewpoint across PMO, enterprise architecture, and service portfolio
- Improves data quality for decision-making
- Provides KPIs and roadmaps to improve business communication

Value Drivers

Holistic Demand
Across PMO, Enterprise Architecture, and service portfolio mgmt.

Business Priorities
Decisions are based on business needs.

Data Consistency
Reliability and trust based on consistent data across services.

Financial Visibility
Information on investment activity and value realization.

Traceability
Link from business request to what was delivered.

Communication
With business stakeholders through service roadmaps.

Proof Points

Innovation — % of new investment vs maintenance
Demand — By source and type
Costs — % planned vs actual
Compliance — Deficiencies in security policies and standards
Capital — % CapEx vs OpEx
Usage — % satisfied customers per service

Key Activities

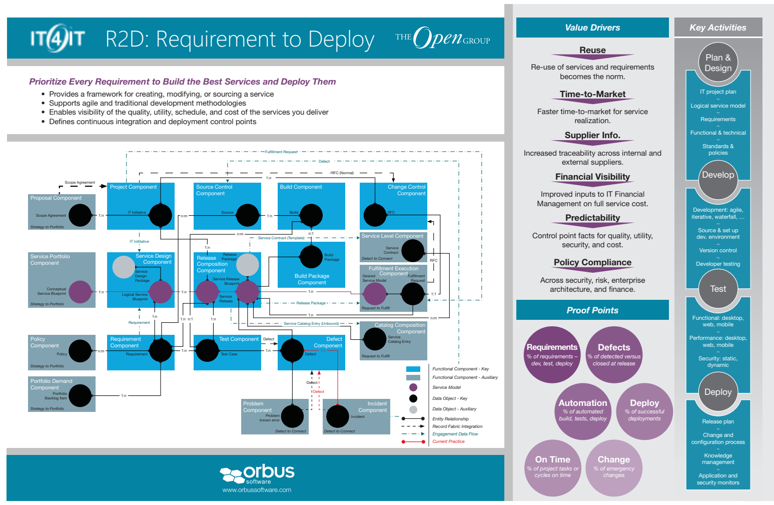
Strategy: Define objectives ~ Align business and IT roadmaps ~ Set up standards and policies

Service Portfolio: Enterprise architecture ~ Service portfolio rationalization ~ Create service blueprint and roadmap

Demand: Consolidate demand ~ Analyze priority, urgency, and impact ~ Create new or tag existing demand

Selection: Business value, risk, costs, benefits, & resources ~ What-if analysis ~ Ensure governance

orbus software
www.orbussoftware.com



IT4IT R2D: Requirement to Deploy

THE Open GROUP

Prioritize Every Requirement to Build the Best Services and Deploy Them

- Provides a framework for creating, modifying, or sourcing a service
- Supports agile and traditional development methodologies
- Enables visibility of the quality, utility, schedule, and cost of the services you deliver
- Defines continuous integration and deployment control points

Value Drivers

Reuse
Re-use of services and requirements becomes the norm.

Time-to-Market
Faster time-to-market for service realization.

Supplier Info.
Increased traceability across internal and external suppliers.

Financial Visibility
Improved inputs to IT Financial Management on full service cost.

Predictability
Control point facts for quality, utility, security, and cost.

Policy Compliance
Across security, risk, enterprise architecture, and finance.

Proof Points

Requirements — % of requirements – dev, test, deploy
Defects — % of detected versus closed at release
Automation — % of automated build, tests, deploy
Deploy — % of successful deployments
On Time — % of project tasks or cycles on time
Change — % of emergency changes

Key Activities

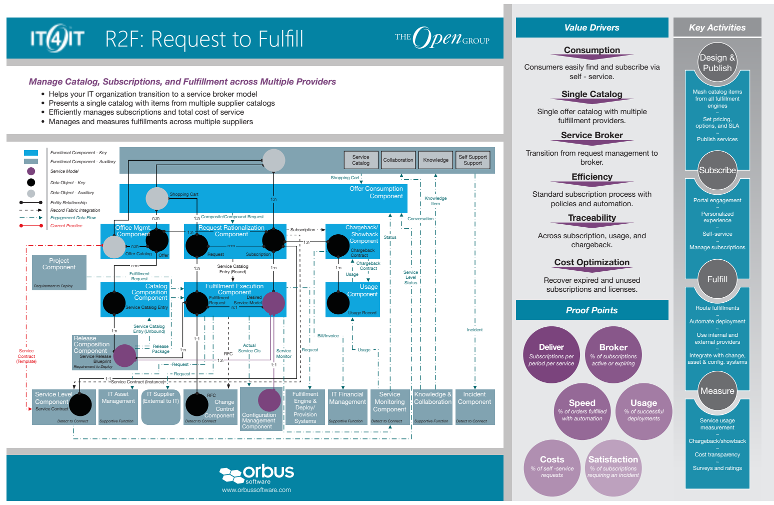
Plan & Design: IT project plan ~ Logical service model ~ Requirements ~ Functional & technical ~ Standards & policies

Develop: Development: agile, iterative, waterfall, … ~ Source & set up dev. environment ~ Version control ~ Developer testing

Test: Functional: desktop, web, mobile ~ Performance: desktop, web, mobile ~ Security: static, dynamic

Deploy: Release plan ~ Change and configuration process ~ Knowledge management ~ Application and security monitors

orbus software
www.orbussoftware.com



IT4IT R2F: Request to Fulfill

THE Open GROUP

Manage Catalog, Subscriptions, and Fulfillment across Multiple Providers

- Helps your IT organization transition to a service broker model
- Presents a single catalog with items from multiple supplier catalogs
- Efficiently manages subscriptions and total cost of service
- Manages and measures fulfillments across multiple suppliers

Value Drivers

Consumption
Consumers easily find and subscribe via self - service.

Single Catalog
Single offer catalog with multiple fulfillment providers.

Service Broker
Transition from request management to broker.

Efficiency
Standard subscription process with policies and automation.

Traceability
Across subscription, usage, and chargeback.

Cost Optimization
Recover expired and unused subscriptions and licenses.

Proof Points

Deliver — Subscriptions per period per service
Broker — % of subscriptions active or expiring
Speed — % of orders fulfilled with automation
Usage — % of successful deployments
Costs — % of self -service requests
Satisfaction — % of subscriptions requiring an incident

Key Activities

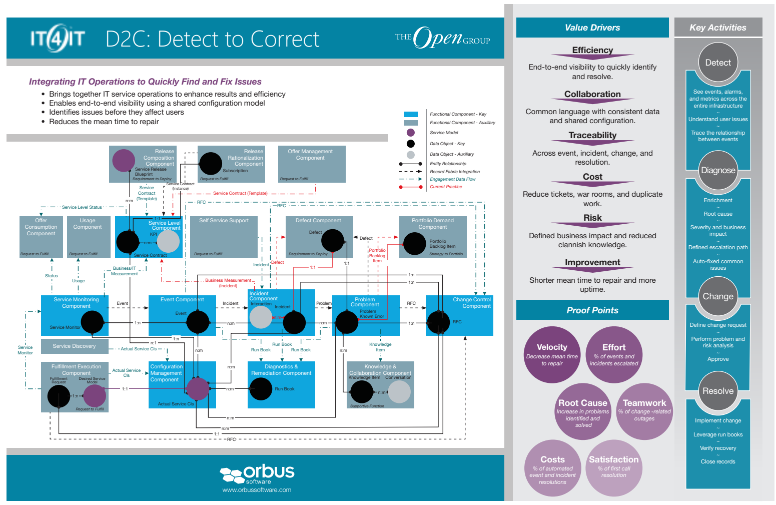
Design & Publish: Mash catalog items from all fulfillment engines ~ Set pricing, options, and SLA ~ Publish services

Subscribe: Portal engagement ~ Personalized experience ~ Self-service ~ Manage subscriptions

Fulfill: Route fulfillments ~ Automate deployment ~ Use internal and external providers ~ Integrate with change, asset & config. systems

Measure: Service usage measurement ~ Chargeback/showback ~ Cost transparency ~ Surveys and ratings

orbus software
www.orbussoftware.com



IT4IT D2C: Detect to Correct

THE Open GROUP

Integrating IT Operations to Quickly Find and Fix Issues

- Brings together IT service operations to enhance results and efficiency
- Enables end-to-end visibility using a shared configuration model
- Identifies issues before they affect users
- Reduces the mean time to repair

Value Drivers

Efficiency
End-to-end visibility to quickly identify and resolve.

Collaboration
Common language with consistent data and shared configuration.

Traceability
Across event, incident, change, and resolution.

Cost
Reduce tickets, war rooms, and duplicate work.

Risk
Defined business impact and reduced clannish knowledge.

Improvement
Shorter mean time to repair and more uptime.

Proof Points

Velocity — Decrease mean time to repair
Effort — % of events and incidents escalated
Root Cause — Increase in problems identified and solved
Teamwork — % of change -related outages
Costs — % of automated event and incident resolutions
Satisfaction — % of first call resolution

Key Activities

Detect: See events, alarms, and metrics across the entire infrastructure ~ Understand user issues ~ Trace the relationship between events

Diagnose: Enrichment ~ Root cause ~ Severity and business impact ~ Defined escalation path ~ Auto-fixed common issues

Change: Define change request ~ Perform problem and risk analysis ~ Approve

Resolve: Implement change ~ Leverage run books ~ Verify recovery ~ Close records

orbus software
www.orbussoftware.com

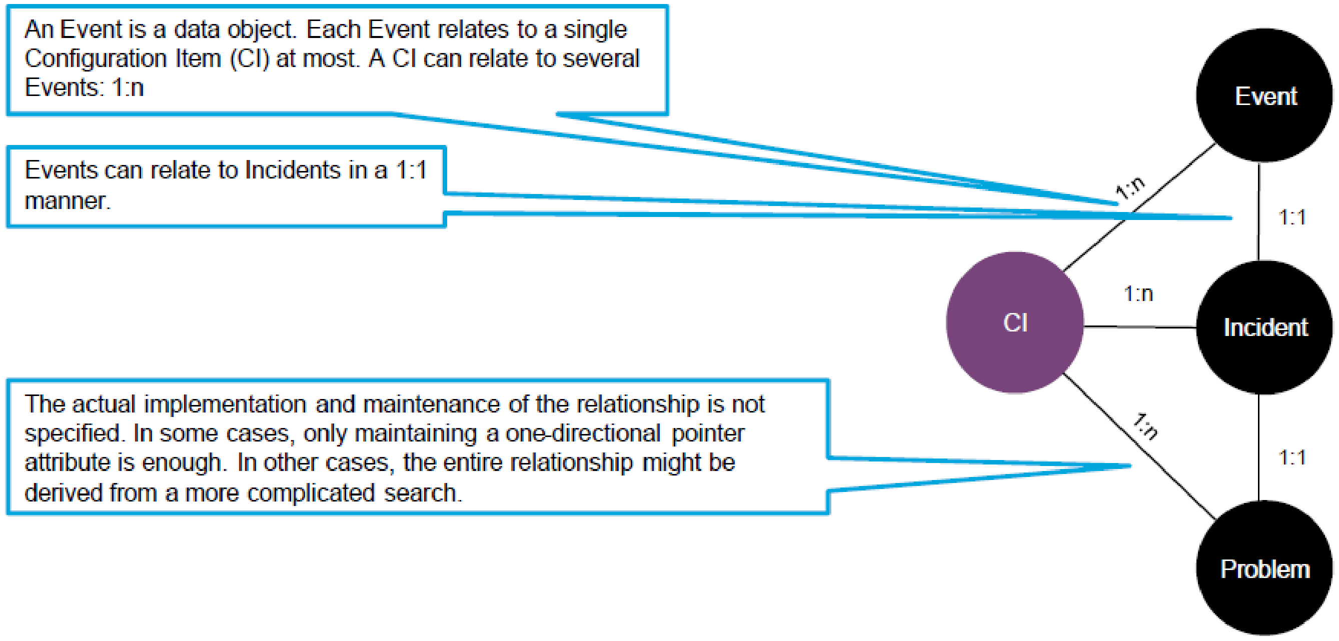
orbus software
www.orbussoftware.com

IT4IT Definitions and Symbols

THE Open GROUP

Information Model

Relationship Type	Description	Informal Representation	Formal Representation
Entity Relationship	The essential relationships between data objects within and across value streams are defined in the IT4IT Reference Architecture. These relationships function as a prescriptive guide for ensuring the integrity of the Service Model as it progresses through its lifecycle and facilitate traceability across value streams. Within the IT4IT Reference Architecture, the relationship between data objects is annotated as follows:	•—1:1—•	0..1 —— 0..1
	1 to 1 (1:1): implies that if there is a relationship, it is between two data objects. It does not imply that there will always be a relationship. For example, Events without Incidents or Incidents without Events are legitimate scenarios.	1:1	0..1:0..1
	1 to many (1:n): implies that one data object relates (A) to one or more other data objects (B...) in scenarios where there is a relationship.	1:n	0..1:*
	Many to many (n:m): implies that both and A and B above can relate to zero, one, or many of the connected data objects.	n:m	*..*:*
Record Fabric Relationship	These entity relationship definitions ensure the consistent management of the lifecycle for individual data objects, as well as ensuring that the data objects are consistently named and crosslinked through prescriptive data flows between functional components to maintain the integrity of the Service Model. They are represented by a dotted black line.	- - - - - - - -►	
Engagement Dataflow	These are user interface integrations derived from value stream use-cases and user stories. These integrations deliver the technology underpinning for a capability by combining several functional components into a single user experience to facilitate human interaction with data objects. In the IT4IT Reference Architecture system of engagement integrations are represented by the blue arrow	—·—·—·►	

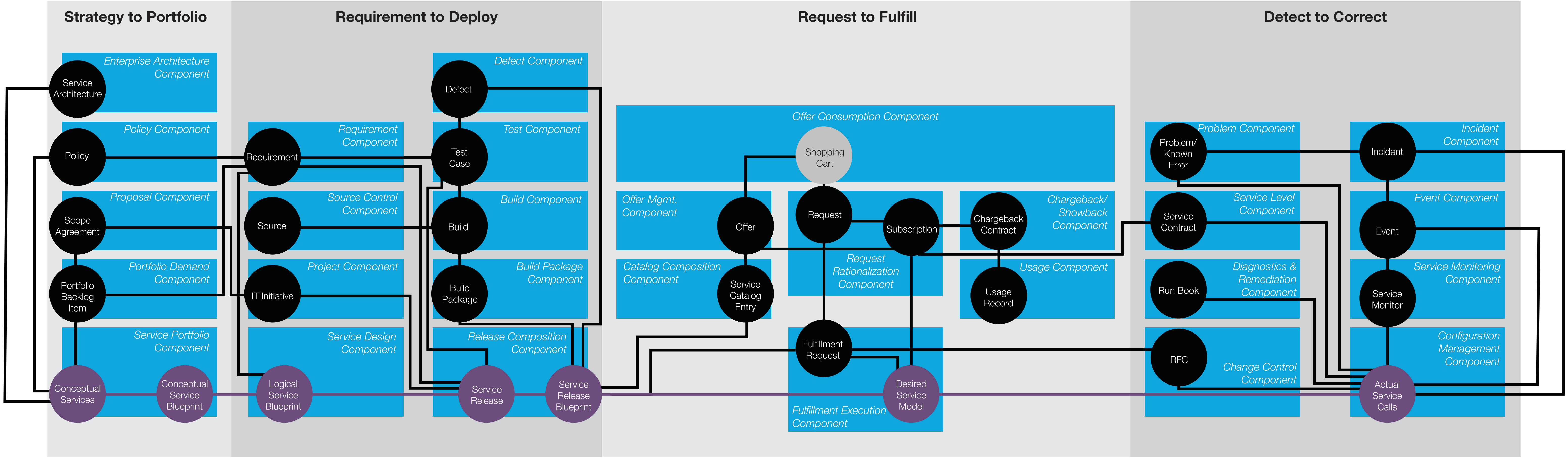


An Event is a data object. Each Event relates to a single Configuration Item (CI) at most. A CI can relate to several Events: 1:n

Events can relate to Incidents in a 1:1 manner.

The actual implementation and maintenance of the relationship is not specified. In some cases, only maintaining a one-directional pointer attribute is enough. In other cases, the entire relationship might be derived from a more complicated search.

IT4IT™ Reference Architecture, Level 1



Strategy to Portfolio

Enterprise Architecture Component

Service Architecture

Policy Component

Policy

Proposal Component

Scope Agreement

Portfolio Demand Component

Portfolio Backlog Item

Service Portfolio Component

Conceptual Services

Conceptual Service Blueprint

Requirement to Deploy

Requirement Component

Requirement

Source Control Component

Source

Project Component

IT Initiative

Service Design Component

Logical Service Blueprint

Defect Component

Defect

Test Component

Test Case

Build Component

Build

Build Package Component

Build Package

Release Composition Component

Service Release

Service Release Blueprint

Request to Fulfill

Offer Consumption Component

Shopping Cart

Offer Mgmt. Component

Offer

Catalog Composition Component

Service Catalog Entry

Request

Subscription

Request Rationalization Component

Chargeback/ Showback Component

Chargeback Contract

Usage Component

Usage Record

Fulfillment Request

Desired Service Model

Fulfillment Execution Component

Detect to Correct

Problem Component

Problem/ Known Error

Service Level Component

Service Contract

Diagnostics & Remediation Component

Run Book

RFC

Change Control Component

Incident Component

Incident

Event Component

Event

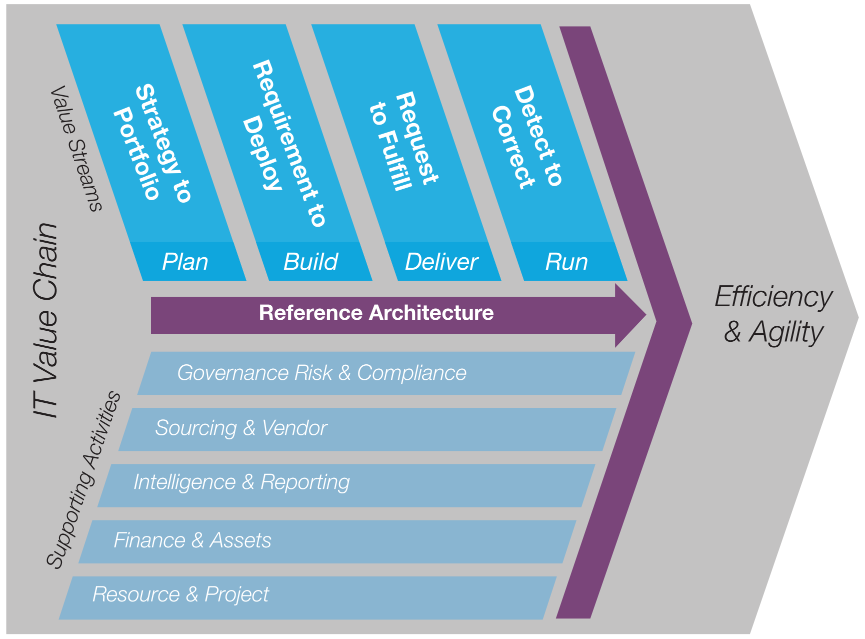
Service Monitoring Component

Service Monitor

Configuration Management Component

Actual Service Calls

Leveraging Business Value Chain Success



IT Value Chain

Value Streams

Strategy to Portfolio

Requirement to Deploy

Request to Fulfill

Detect to Correct

Plan

Build

Deliver

Run

Reference Architecture

Supporting Activities

Governance Risk & Compliance

Sourcing & Vendor

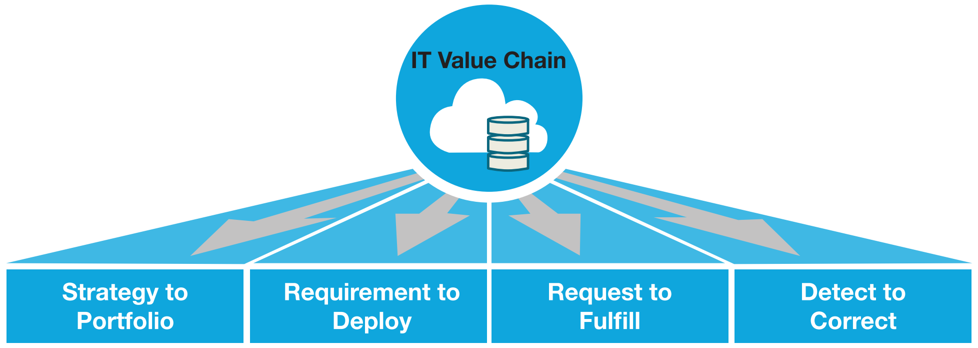
Intelligence & Reporting

Finance & Assets

Resource & Project

Efficiency & Agility

Value Stream Overview



IT Value Chain

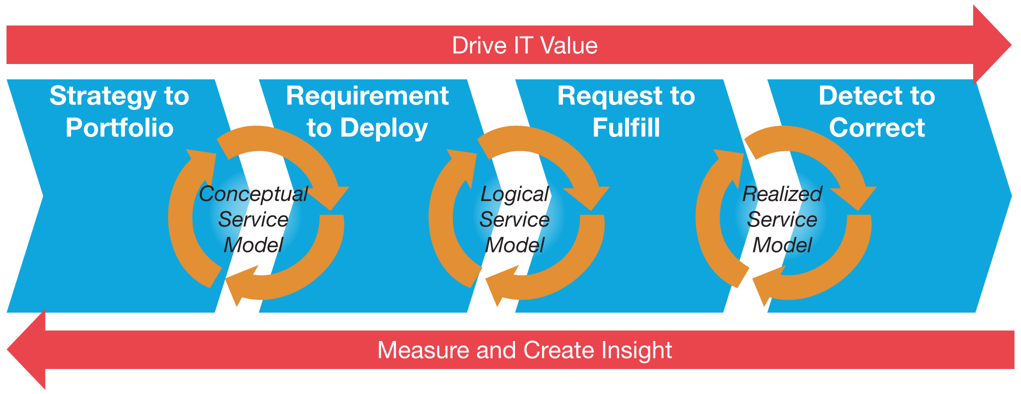
Strategy to Portfolio

Requirement to Deploy

Request to Fulfill

Detect to Correct

Service Model Lifestyle



Drive IT Value

Strategy to Portfolio

Requirement to Deploy

Request to Fulfill

Detect to Correct

Conceptual Service Model

Logical Service Model

Realized Service Model

Measure and Create Insight

An Operating Model for the New Style of IT

Plan Define your **strategy** to balance and broker your **Portfolio**

Build Prioritize every **requirement** to build the best services and **deploy** them

Deliver Handle each **request** by streamlining the process to **fulfill** it

Run Seek to **detect** issues and to **correct** them before impacting users

End-to-End Integrated Reference Architecture

Portfolio Mgmt.

Cloud Automation

Application Lifecycle Mgmt.

Application Mobilization

Automation and Orchestration

IT Service Mgmt.

Business Service Mgmt.

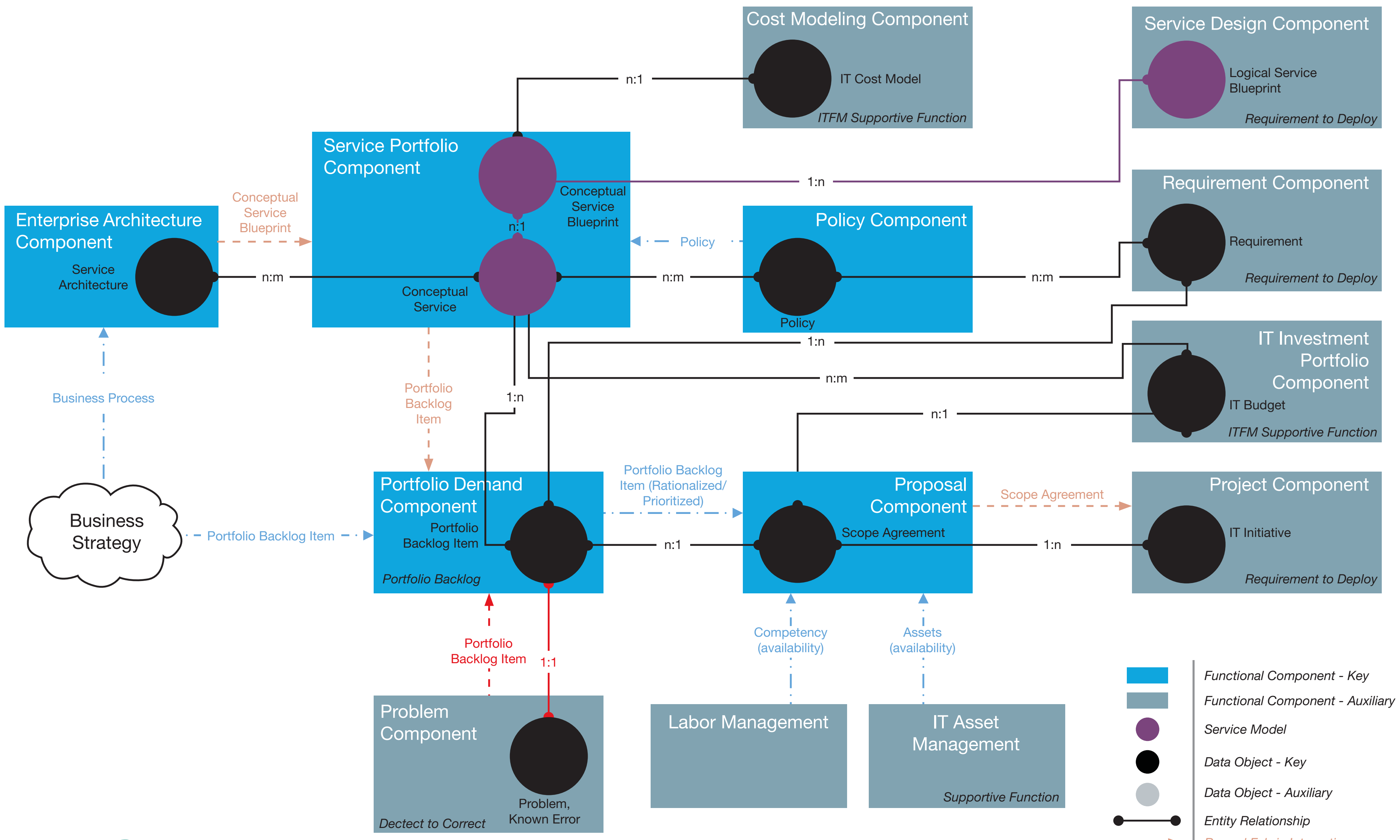
Source: Open Group - http://www.opengroup.org/IT4IT

orbus software
www.orbussoftware.com

IT4IT™ S2P: Strategy to Portfolio

Manage your IT Portfolio and Investments to Drive Business Innovation

- Provides the strategy to balance and broker your portfolio
- Provides a unified viewpoint across PMO, enterprise architecture, and service portfolio
- Improves data quality for decision-making
- Provides KPIs and roadmaps to improve business communication



Cost Modeling Component
IT Cost Model
ITFM Supportive Function

Service Design Component
Logical Service Blueprint
Requirement to Deploy

Service Portfolio Component
Conceptual Service Blueprint
Conceptual Service

Requirement Component
Requirement
Requirement to Deploy

Enterprise Architecture Component
Service Architecture

Policy Component
Policy

IT Investment Portfolio Component
IT Budget
ITFM Supportive Function

Portfolio Demand Component
Portfolio Backlog Item
Portfolio Backlog

Proposal Component
Scope Agreement

Project Component
IT Initiative
Requirement to Deploy

Business Strategy

Problem Component
Problem, Known Error
Dectect to Correct

Labor Management

IT Asset Management
Supportive Function

n:1 · 1:n · n:m · n:1 · 1:n · n:m · 1:n · n:m · n:1 · n:1 · 1:n · 1:1

Conceptual Service Blueprint
Policy
Business Process
Portfolio Backlog Item
Portfolio Backlog Item
Portfolio Backlog Item (Rationalized/ Prioritized)
Scope Agreement
Portfolio Backlog Item
Competency (availability)
Assets (availability)

- 🟦 *Functional Component - Key*
- ▨ *Functional Component - Auxiliary*
- ● *Service Model*
- ● *Data Object - Key*
- ○ *Data Object - Auxiliary*
- ●—● *Entity Relationship*
- - - → *Record Fabric Integration*
- -·-→ *Engagement Data Flow*
- ●—● *Current Practice*

THE Open GROUP

Source: Open Group - http://www.opengroup.org/IT4IT

orbus software
www.orbussoftware.com

Value Drivers

Holistic Demand

Across PMO, Enterprise Architecture, and service portfolio mgmt.

Business Priorities

Decisions are based on business needs.

Data Consistency

Reliability and trust based on consistent data across services.

Financial Visibility

Information on investment activity and value realization.

Traceability

Link from business request to what was delivered.

Communication

With business stakeholders through service roadmaps.

Proof Points

Innovation
% of new investment vs maintenance

Demand
By source and type

Costs
% planned vs actual

Compliance
Deficiencies in security policies and standards

Capital
% CapEx vs OpEx

Usage
% satisfied customers per service

Key Activities

Strategy

Define objectives
~
Align business and IT roadmaps
~
Set up standards and policies

Service Portfolio

Enterprise architecture
~
Service portfolio rationalization
~
Create service blueprint and roadmap

Demand

Consolidate demand
~
Analyze priority, urgency, and impact
~
Create new or tag existing demand

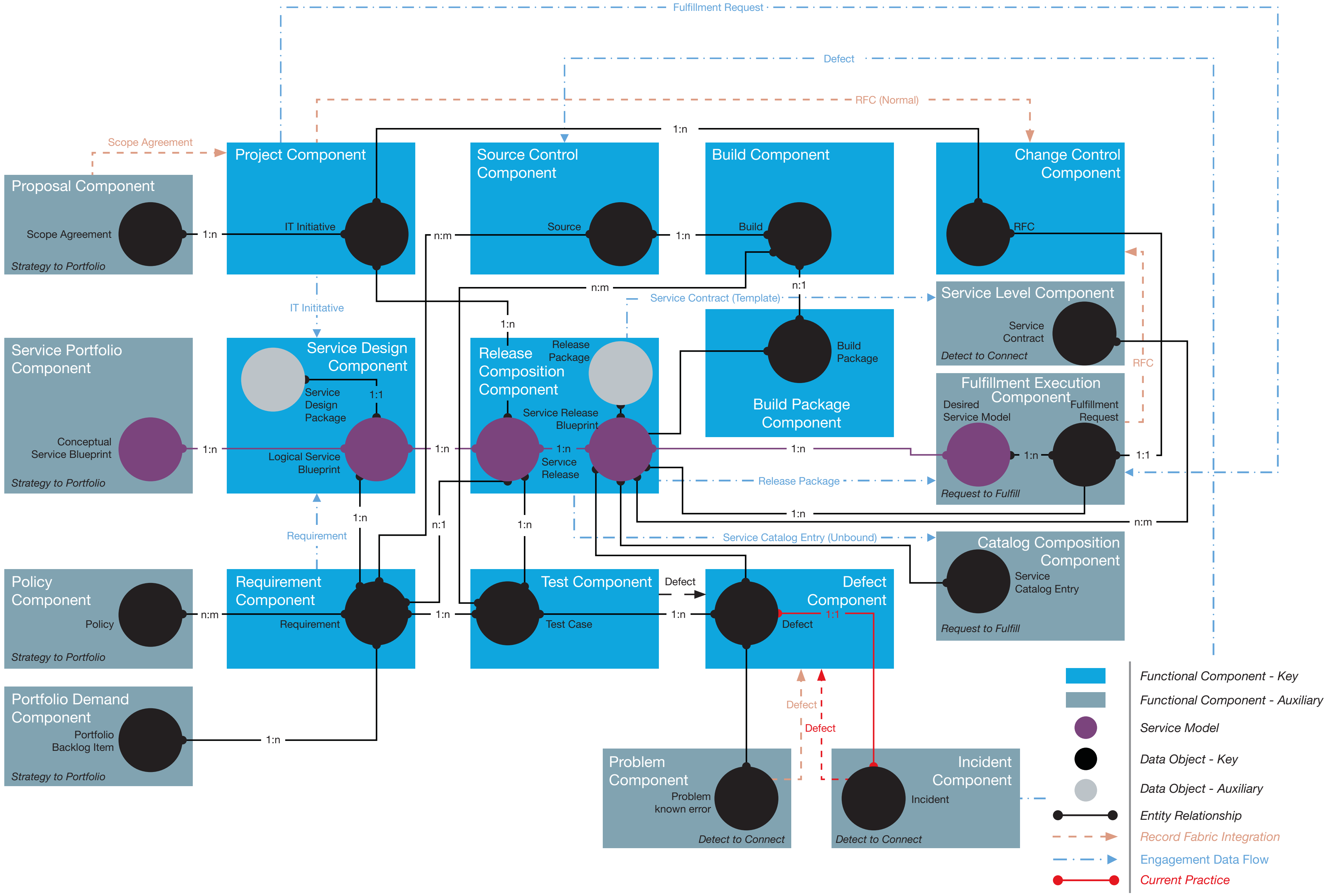
Selection

Business value, risk, costs, benefits, & resources
~
What-if analysis
~
Ensure governance

IT4IT™ R2D: Requirement to Deploy

Prioritize Every Requirement to Build the Best Services and Deploy Them

- Provides a framework for creating, modifying, or sourcing a service
- Supports agile and traditional development methodologies
- Enables visibility of the quality, utility, schedule, and cost of the services you deliver
- Defines continuous integration and deployment control points



Source: Open Group - http://www.opengroup.org/IT4IT

Value Drivers

Reuse

Re-use of services and requirements becomes the norm.

Time-to-Market

Faster time-to-market for service realization.

Supplier Info.

Increased traceability across internal and external suppliers.

Financial Visibility

Improved inputs to IT Financial Management on full service cost.

Predictability

Control point facts for quality, utility, security, and cost.

Policy Compliance

Across security, risk, enterprise architecture, and finance.

Proof Points

Requirements
% of requirements – dev, test, deploy

Defects
% of detected versus closed at release

Automation
% of automated build, tests, deploy

Deploy
% of successful deployments

On Time
% of project tasks or cycles on time

Change
% of emergency changes

Key Activities

Plan & Design

IT project plan
~
Logical service model
~
Requirements
~
Functional & technical
~
Standards & policies

Develop

Development: agile, iterative, waterfall, …
~
Source & set up dev. environment
~
Version control
~
Developer testing

Test

Functional: desktop, web, mobile
~
Performance: desktop, web, mobile
~
Security: static, dynamic

Deploy

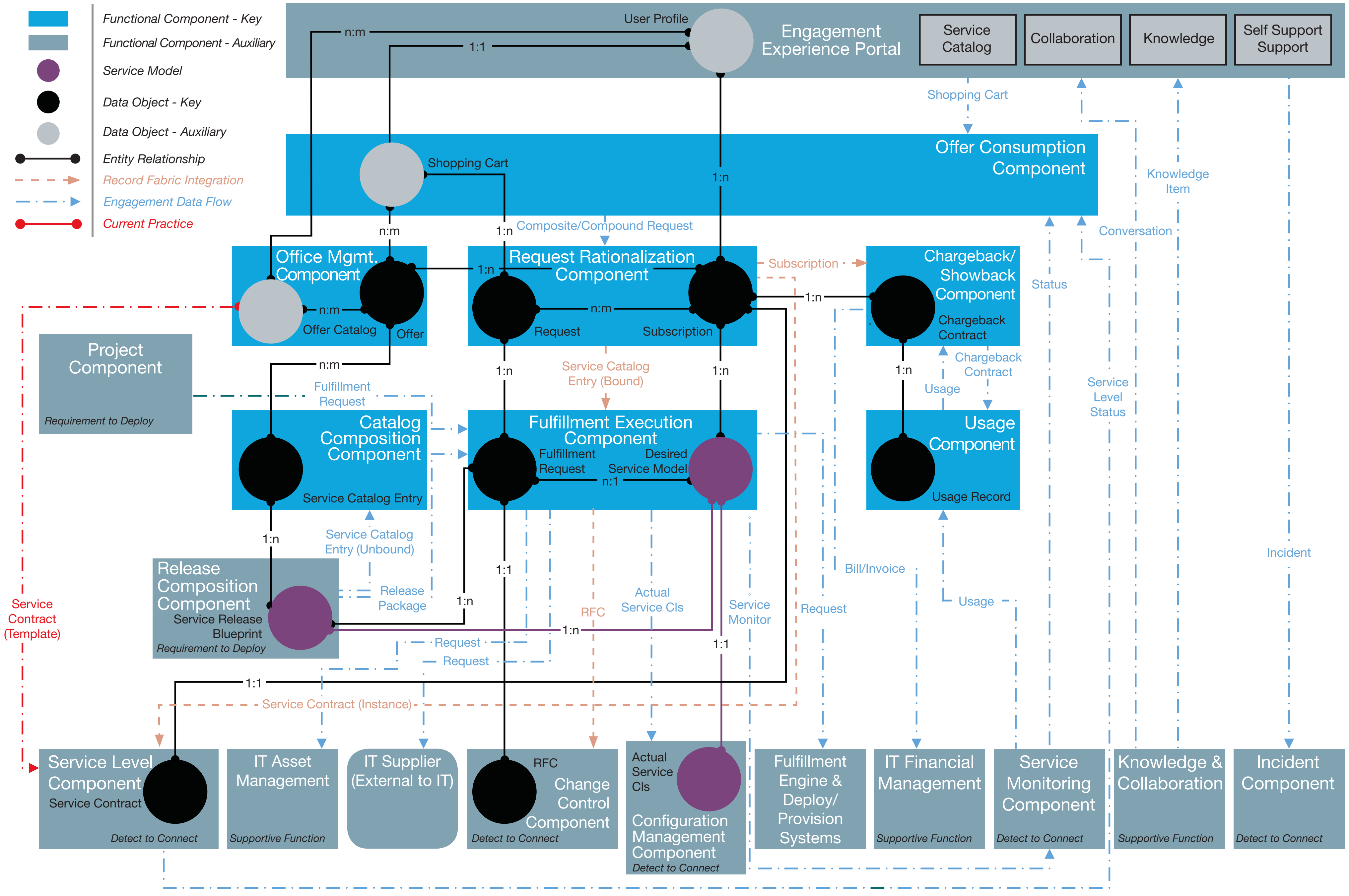
Release plan
~
Change and configuration process
~
Knowledge management
~
Application and security monitors

orbus software
www.orbussoftware.com

IT4IT™ R2F: Request to Fulfill

Manage Catalog, Subscriptions, and Fulfillment across Multiple Providers

- Helps your IT organization transition to a service broker model
- Presents a single catalog with items from multiple supplier catalogs
- Efficiently manages subscriptions and total cost of service
- Manages and measures fulfillments across multiple suppliers



Source: Open Group - http://www.opengroup.org/IT4IT

Value Drivers

Consumption

Consumers easily find and subscribe via self - service.

Single Catalog

Single offer catalog with multiple fulfillment providers.

Service Broker

Transition from request management to broker.

Efficiency

Standard subscription process with policies and automation.

Traceability

Across subscription, usage, and chargeback.

Cost Optimization

Recover expired and unused subscriptions and licenses.

Proof Points

Deliver
Subscriptions per period per service

Broker
% of subscriptions active or expiring

Speed
% of orders fulfilled with automation

Usage
% of successful deployments

Costs
% of self -service requests

Satisfaction
% of subscriptions requiring an incident

Key Activities

Design & Publish

Mash catalog items from all fulfillment engines
~
Set pricing, options, and SLA
~
Publish services

Subscribe

Portal engagement
~
Personalized experience
~
Self-service
~
Manage subscriptions

Fulfill

Route fulfillments
~
Automate deployment
~
Use internal and external providers
~
Integrate with change, asset & config. systems

Measure

Service usage measurement
~
Chargeback/showback
~
Cost transparency
~
Surveys and ratings

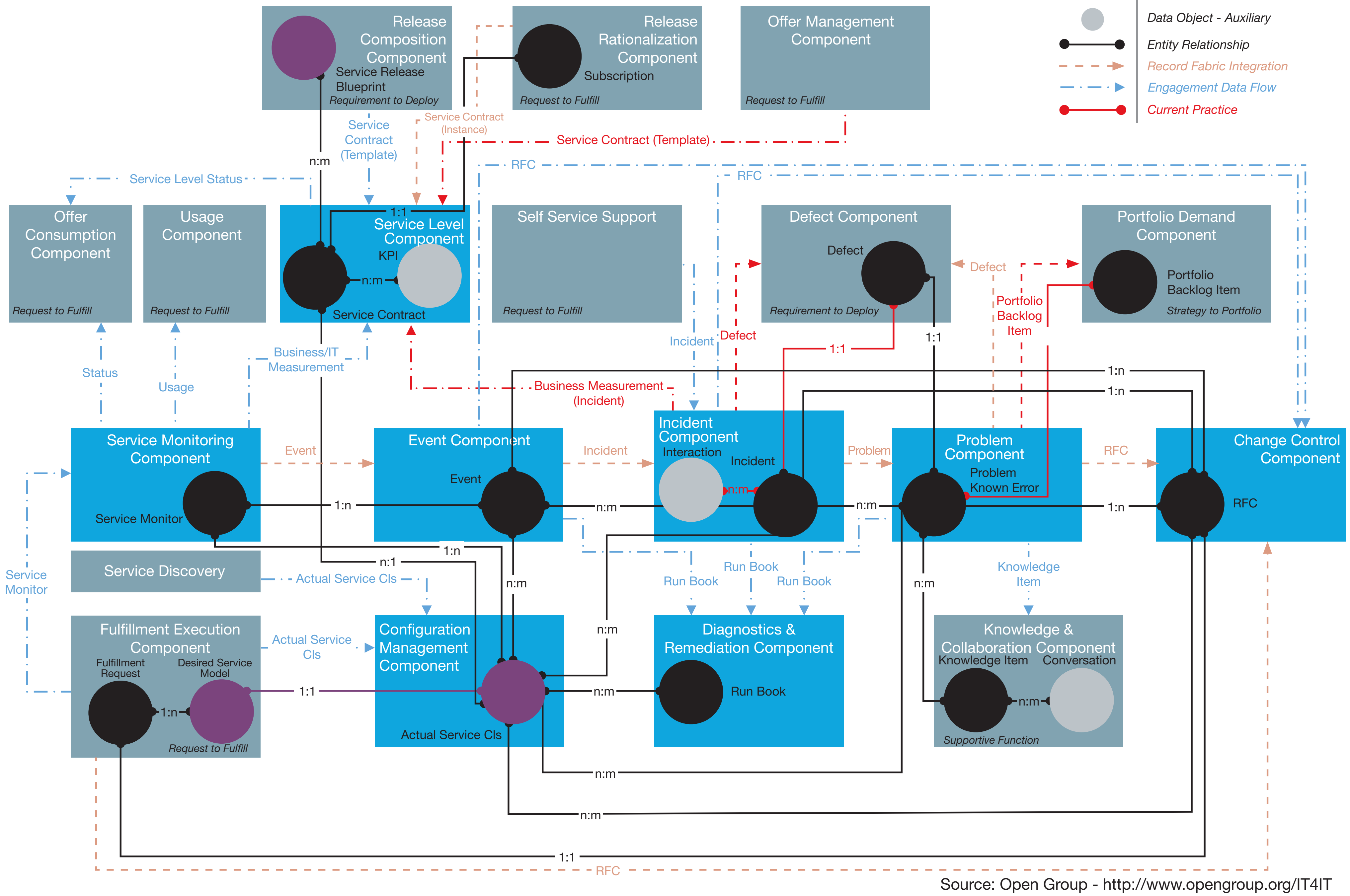
orbus software
www.orbussoftware.com

IT4IT D2C: Detect to Correct

Integrating IT Operations to Quickly Find and Fix Issues

- Brings together IT service operations to enhance results and efficiency
- Enables end-to-end visibility using a shared configuration model
- Identifies issues before they affect users
- Reduces the mean time to repair

🟦	*Functional Component - Key*
▨	*Functional Component - Auxiliary*
●	*Service Model*
●	*Data Object - Key*
○	*Data Object - Auxiliary*
●—●	*Entity Relationship*
- - →	*Record Fabric Integration*
-·-→	*Engagement Data Flow*
●—●	*Current Practice*



Source: Open Group - http://www.opengroup.org/IT4IT

Value Drivers

Efficiency

End-to-end visibility to quickly identify and resolve.

Collaboration

Common language with consistent data and shared configuration.

Traceability

Across event, incident, change, and resolution.

Cost

Reduce tickets, war rooms, and duplicate work.

Risk

Defined business impact and reduced clannish knowledge.

Improvement

Shorter mean time to repair and more uptime.

Proof Points

Velocity
Decrease mean time to repair

Effort
% of events and incidents escalated

Root Cause
Increase in problems identified and solved

Teamwork
% of change -related outages

Costs
% of automated event and incident resolutions

Satisfaction
% of first call resolution

Key Activities

Detect

See events, alarms, and metrics across the entire infrastructure
~
Understand user issues
~
Trace the relationship between events

Diagnoze

Enrichment
~
Root cause
~
Severity and business impact
~
Defined escalation path
~
Auto-fixed common issues

Change

Define change request
~
Perform problem and risk analysis
~
Approve

Resolve

Implement change
~
Leverage run books
~
Verify recovery
~
Close records

orbus software
www.orbussoftware.com