intelligent Workload Distribution

Overview

Copyright

Disclaimer

Feedback

Genesys endeavors to provide accurate and useful documentation for all Genesys products. To achieve this goal, the documentation group welcomes your comments and suggestions regarding any aspect of Genesys user documentation. Send your comments by e-mail to: techpubs.webadmin@genesyslab.com.

Trademarks and Registered Trademarks

Revision History

Version	Issued	Description
1.0	18.06.2007	Initial GTL release 1.0
1.1	01.10.2007	GTL release 1.1
1.2	16.11.2007	GTL release 1.2
7.6.1.1	14.04.2009	iWD release 7.6.1

Contents

Preface

Welcome to the *intelligent Workload Distribution Overview*. This document introduces you to the Genesys intelligent Workload Distribution *(iWD)* and its main functions.

Intended Audience

This document has three primary audiences:

- IT staff who are responsible for the iWD installation and system configuration (such as servers, connections, and services).
- Business analysts who are responsible for iWD business configuration (such as contracts, business processes, and rules).
- Team managers who are responsible for managing enterprise resources who are working on tasks that are supplied by iWD.

The first chapter "intelligent Workload Distribution Features," provides an overview of iWD from a functional perspective and is intended for all audiences.

Second chapter "intelligent Workload Distribution Architecture" describes technical aspects that enable iWD functionality and is primarily intended for IT personnel who will be installing and performing iWD system configuration.

Related Resources

See the following additional resources, for more information:

- *iWD Quick Start Guide*
- *iWD Deployment Guide*
- *iWD Manager Guide*

Document Conventions

This document uses the following stylistic and typographical conventions, which serve to identify specific types of information:

Type Styles

Italic

In this document, italic text denotes emphasis, document titles, definitions of (or first references to) unfamiliar terms, and mathematical variables.

For example:

- Please consult the *intelligent Workload Distribution Manager User Guide* for more information.
- *Do not use* this value for this option.
- The formula, $x+1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which resembles `teletype` or `typewriter` text, is used for all programming identifiers and graphical user interface (GUI) elements.

This convention includes the names of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

For example:

- Select the `Default` check box.
- Click the `Edit` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- Click `OK` to exit the `Properties` dialog box.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

For example:

- Enter `exit` on the command line.

intelligent Workload Distribution Features

intelligent Workload Distribution (iWD) is an enterprise-wide task list that is centrally managed and prioritized. It enables work to be presented to the right resource, at the right time, and at the right location. It gets non-real-time work (tasks) from the Web, uses business rules to prioritize or reprioritize, and then distributes the tasks to the most suitable resource (performer).

Capturing Tasks from Multiple Sources

A key function of the iWD solution is the ability to capture - or get - work from the multitude of work sources in the enterprise. The iWD accepts work, in electronic format, from the following types of systems:

- *Electronic Web Forms* that are present on either Internet or intranet websites in the enterprise. Values such as customer-identification number that are entered into the form can be more easily validated. Forms can be specific to a process such as an order or request for a refund, or to ageneral request.
- *E-Mail Management systems* that capture and analyze e-mails that provide a final source-of-work capture point for the iWD, in which e-mails can be prioritized with voice calls and with all non-real-time tasks in the enterprise.
- *Document Management and Fax systems* through Capture Adapters that are designed for leading third-party systems.
- *Databases* for custom applications.
- *Web Services* for custom applications that have service-oriented interfaces.

Capture Points are enabled by the Adapter. Each Capture Point is the configuration of the Adapter for capturing a specific sort of work - often, associated with a specific business process, such as an order, refund, or return. The Adapter is technology-specific, while the Capture Point is specific to a business process.

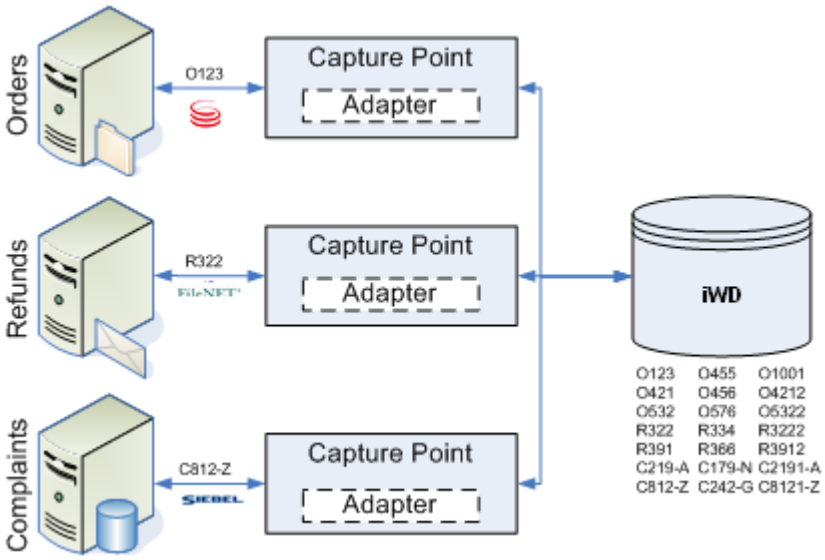

Figure 1: Capture Point process diagram

To establish a connection with the correct target system and the corresponding source of work, each Capture Point requires the configuration of properties such as web form and workflow identifiers.

Prioritization of Tasks

Having a consolidated set of tasks is only the first step in achieving the business benefits of increased efficiency and effectiveness. As with any business, the fulfilment of one task over another often results in a benefit (such as increased revenue, decreased cost, or avoidance of a penalty, fine, or loss of customer satisfaction). Prioritization is the process by which iWD arranges the task list in order of priority or importance, based on business rules that are configured for the Capture Point, Process, and Contract.

Business rules within iWD are based on Business-Rule Templates that are created by IT personnel. As with templates for Microsoft Office Word, business-rule templates are the foundation upon which business users create or configure existing business rules that govern iWD - thereby, empowering business users to make the necessary changes, when required, while allowing IT personnel to focus on IT-related tasks.

Business rules can also be used to facilitate a more informed prioritization decision in leveraging existing enterprise web services to collect additional task-related information. For example, you can configure a rule that, on capture for any Order, retrieves the customer's current credit score. The score (such as poor) is then attached to the task and can be utilized downstream by another business rule for setting priority, or later by the router itself for determining to whom the task should be routed (for example, credit-granting department prior to sales).

Reprioritization of Tasks

At any time, the information that is related to a task can change and affect the task's priority. A simple example of where reprioritization can affect the initial priority set is the time that remains before the due date for a task. You can configure a business rule, such that if the Interval Time Remaining for a task that is associated to a time-sensitive process - for example, a dispute where (if the dispute is not completed within 10 days) the organization may be fined - is fewer than two days, then assign it the highest priority, for immediate assignment to an agent.

The iWD is configured where each captured task receives a task reprioritization interval, when business rules are applied and new values are set for the task. Some might increase in priority, while others might decrease.

All business rules have a unique identifier and description, such that any rule that is applied and the value that is applied are captured and stored for future business reporting, including business and internal auditing.

Task Distribution

Distribution Points are similar to Capture Points; however, instead of defining where to retrieve tasks, they define the location to which iWD should send tasks for processing. In most cases, these are associated with contact centers, and iWD supports integration with Genesys via the Open Media® interface through a Genesys Open Media ® Distribution Adapter. The configuration of parameters is specific to that interface - for example, the names for the Open Media Inbound and Outbound queues.

While calculating values for priority and applying rules for distribution, iWD presents work to the routing platforms in batches. In most cases, the size of the batch – based on the distribution threshold – is set to a point in which tasks that have been sent to the Interaction Server are completed in a unit of time, typically, measured in hours, whereas the iWD can manage tasks over the course of days or weeks.

The iWD supports multiple Distribution Points to further enhance the work distribution - providing an enterprise to route lower-value tasks selectively (as determined by the capture rules) to a lower-cost region, including routing to a third-party business process or call-center outsourcers.

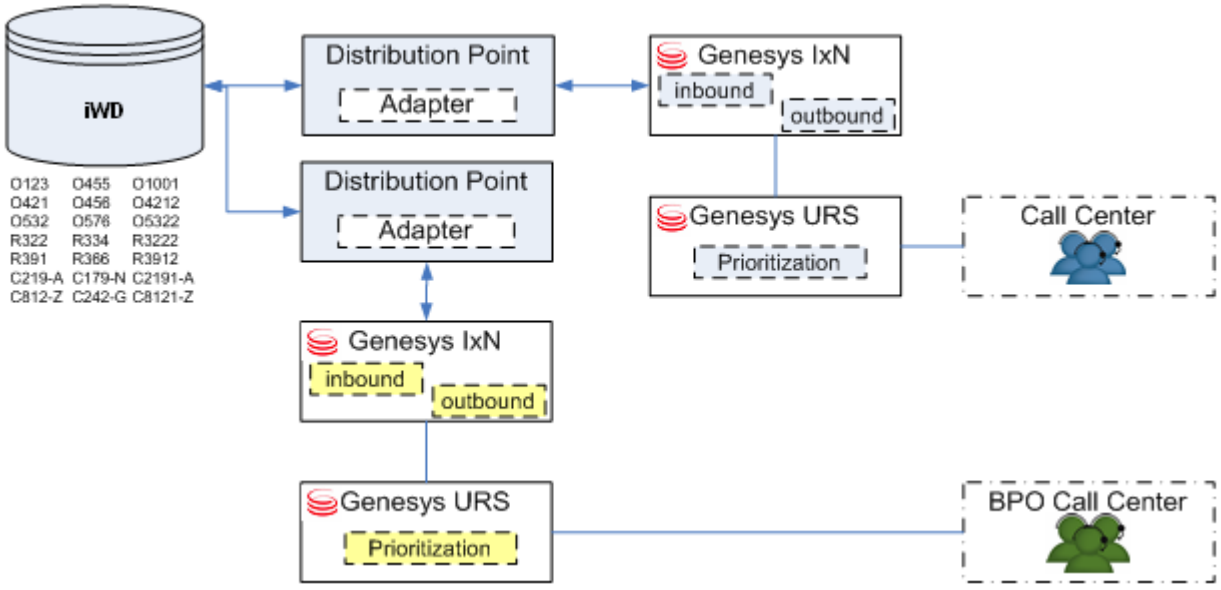

Figure 2: Diagram with multiple Distribution Points

Although the iWD performs prioritization and reprioritization, it does so only to set values for priority routing within the routing server. Using Genesys Open Media® as an example, Genesys Universal Routing Server (URS) can leverage the iWD-calculated priority and business values in its routing strategies, or it might calculate its own - in either case, ensuring that the most critical tasks are presented first to agents. Genesys URS continues to reevaluate priority for tasks that it has received against real-time voice and chat events - ensuring that the most important is presented next. By using bidirectional messaging and integration with the Genesys Statistics Server, iWD ensures that a continuous stream of non-real-time work is presented to the Distribution Point, without jeopardizing service levels.

Distribution of tasks to Genesys by the iWD is not a "fire-and-forget" implementation. iWD maintains an association with the tasks that have been submitted to Genesys in support of the following events:

- Revocation: Used when the priority of a task requires that it be returned to the list given a lower priority, or when an external actor (system or actor) has requested control of the task, or a cancellation of the fulfilment by a contact-center agent.
- Return: When a Task has been assigned but must be returned to the iWD. An example would be when a task had been captured without any supporting data, such as business process or customer identification, and the agent has retrieved this information and updated the task so that iWD may prioritize the task and re-present it again by using the necessary values.
- Updates: When tasks are assigned to the Interaction Server, so that any updates to the task, including reprioritization, are sent to Genesys for the update.

Cradle-to-Grave Reporting

As the central point between capture and distribution, iWD monitors the tasks throughout its life cycle. iWD provides business-activity monitoring of a number of events, including:

- Capture. The point at which the event was captured in the Capture Point. iWD assigns a unique identifier to the task and maintains the association to other task identifiers that are assigned by the capture and distribution system.
- Process/Reprioritization Rules Applied. All process rules that are applied to the task, as well as the values that are calculated and assigned to the task.
- Queued. When the task was ready for presentment to the Genesys Interaction Server.
- Distributed and Assigned. When it was sent to Genesys Interaction Server and assigned to an agent for processing.
- Transfer/Transfer to Queue. Whether the agent should transfer the task to another agent or back to the queue.
- Held/Revoked. Whether the task is held (manual hold) or revoked.

Each record is time-stamped and stored in the iWD reporting domain, where the data that is collected can be leveraged in third-party reporting applications.

intelligent Workload Distribution Architecture

"Things should be made as simple as possible, but not any simpler."

This Albert Einstein quote very precisely characterizes the philosophy behind intelligent Workload Distribution architecture, system design and actual implementation. Nowadays, many if not most solutions suffer from over-complexity. This makes enhancements, upgrade, and integration efforts quite painful and, often, not cost-effective (that is, an enhancement would cost more than the return from it).

During the design and building of the intelligent Workload Distribution, this was one primary focus. The solution components should be just like Lego bricks for software: easily composed into the final solution. Of course, this also means that the architecture should maintain a high level of flexibility, in which many different combinations of building blocks are supported.

Architecture Principles

intelligent Workload Distribution is built from the ground up, following the service-oriented architecture principles. This means that the iWD solution is composed of loosely coupled services, each of which has clean and simple interfaces. To ensure a high level of openness and platform independence, integration-wise, all iWD services are exposed as secure Web Services; hence, iWD can be easily integrated into any system and environment.

Another important iWD architecture principle is an ability to plug-in extensions easily. For this, the solution provides a number of extension points, each of which has a predefined purpose and interface. These extension points can be used to enhance iWD and add new functionality - for example:

- New ways of capturing tasks.
- Additional audit data targets such as data warehouses.
- Additional configuration objects.

To ensure that it is actually easy to set up and combine different solution building blocks, iWD also provides highly optimized configuration capabilities, based on such principles such as:

- DRY (Don't Repeat Yourself). No configuration should be performed twice.
- Maximize automation (for example, provide lookup values wherever possible).

- Minimize necessary scripting.

High-Level Architecture

Given the current climate of the IT industry, the natural choice of implementation platform for iWD was Java. Its wide spread use, reasonable platform independence, and great number of available applications and libraries makes it the preferred choice for most of today's enterprises.

The core iWD solution building blocks are implemented as light Java services, which can run in any modern Java application server, such as Tomcat, JBoss, Weblogic, and Websphere. iWD does not rely on any J2EE-specific functions. In runtime, the services are wired through dependency injection (also know as IoC: Inversion of Control).

Wiring, configuration, and other service runtime management functions are enabled through the Spring open-source framework, which has received wide acceptance in the Java community and has proven itself during many years of use in different enterprise solutions.

The following diagram illustrates the main intelligent Workload Distribution building blocks (services), the relationships among them, and the involved external components:

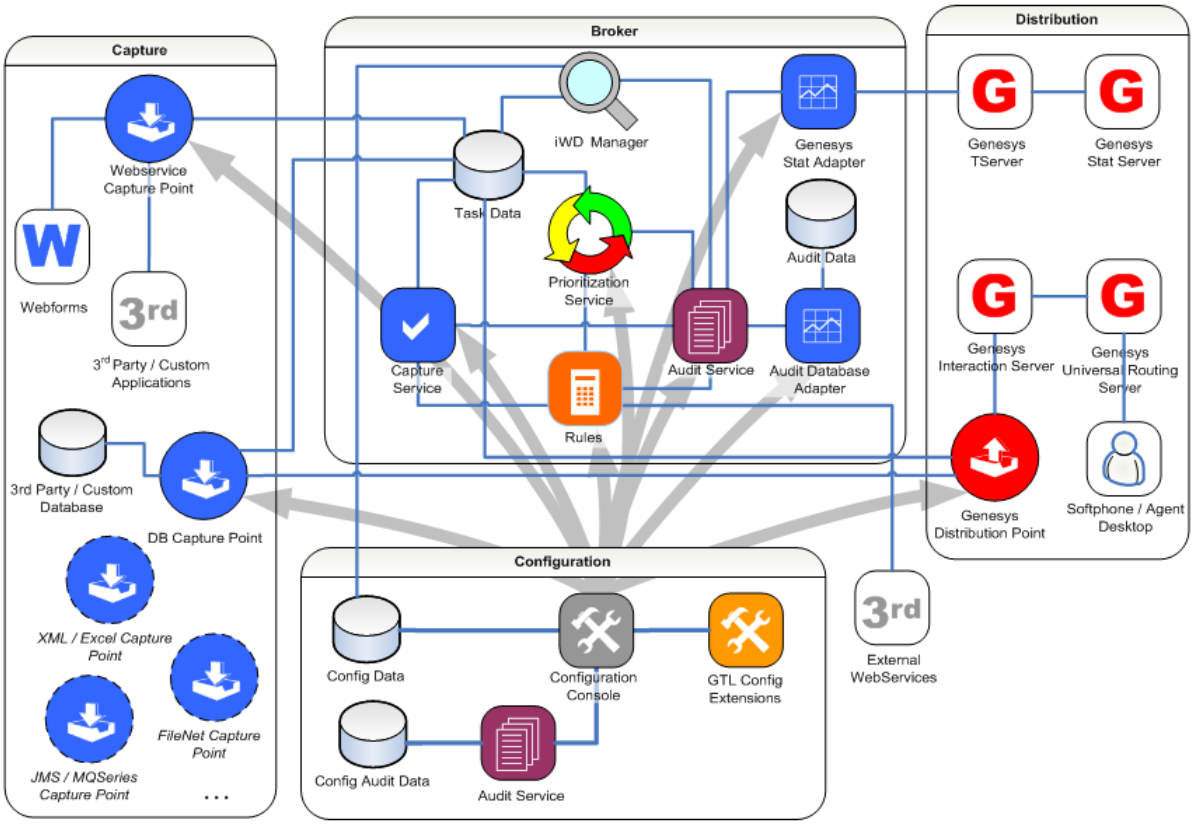

Figure 3: intelligent Workload Distribution high-level architecture

The iWD solution is split into four functional areas, each of which has its own distinctive role:

- Capture: The capture of tasks from different sources

- Broker: The classification, prioritization, and monitoring of tasks
- Distribution: The distribution of tasks for blending with real-time interactions
- Configuration: The configuration for the three other functional areas

Capture

The primary role of the capture functional area is the feeding of new tasks (also known as non-real-time interactions) from different enterprise systems into iWD. Additionally the capture functional area provides capabilities such as:

- Updating submitted tasks.
- Canceling submitted tasks.
- Handling tasks status-change notifications.

Task capture is realized by a special type of iWD service that is called a capture point. There is a number of standard capture points, that can addresses many different task-capture scenarios:

- Webservices Capture Point
- DB Capture Point
- XML/Excel Capture Point
- JMS/Websphere MQ Capture Point

Besides that, it is possible also to implement new custom capture points, as this is one of the extension points that the iWD provides.

Classification and Prioritization

Task classification is a procedure that associates captured tasks with one of the configured business processes. In addition, some of the initial values of the task can be set - for example, task due date, business value of the task, and the date when task priority should be reevaluated by the iWD. Prioritization is a procedure that can continuously adjust the priority of a task, depending on the time that remains before the due date of a task or some other time-related factor.

Classification Service

Task classification is performed by the Classification Service, which executes a set of business rules against the task. These rules are called Classification Rules and are associated with the capture point that captured the task.

Prioritization Service

Task prioritization is performed by the Prioritization Service, which executes a set of business rules against the task. These rules are called Prioritization Rules and are associated with the Contract and/or Process of the task.

Rules

Rules are the main means of configuring the task-processing logic in the iWD. The main essence of rules is an ability to configure complex logic in a simple, nontechnical, user-friendly way. This is achieved by encapsulating an implementation logic into rule templates and exposing this logic to an end user in simple human-language expressions. The implementation logic is first created by IT professionals and put into rule templates which basically represent a set of reusable rule conditions and actions. Nontechnical users can then use these conditions and actions to set up actual rules that describe the business logic; no implementation logic is directly involved.

Execution of rules is performed via JBoss Rules (Drools), which is an extensive and powerful open-source rules engine.

Distribution

Distribution enables submission of tasks from the iWD to Genesys, for blending with real-time interactions and routing to agents. Task submission is realized through a special iWD service type that is called a Distribution Point.

Naturally, tasks are submitted based on prioritization order in the iWD. When a task has been assigned or completed, the distribution point also receives a notification, updates task status in the iWD accordingly, and passes notification back to the capture point that originated the task.

The interface to Genesys is implemented by using the Genesys Platform SDK - specifically, the OpenMedia API.

Auditing

The audit service records all of the events that occur in relation to each task in the iWD, including the details that are related to the event. A few examples of events that are recorded by the audit service are the following:

- Task Assigned: Also logs the user who received the task
- Task Completed
- Rule Applied: Logs what rule and what attributes of a task were changed

The audit service can have a number of adapters that store the audit event. There are two standard adapter, one for recording events into a database (mandatory) and one for submitting events to Genesys Stat Server to enable unified reporting in Genesys (optional). The audit adapter is another extension point that enables support for additional custom adapters.

Configuration

iWD provides a single-point of configuration in a thin-client GUI application that is called the Configuration Console. Any and all configuration that is necessary for the iWD can be performed via this application.

The configuration is stored and managed in centralized database; however, runtime services work with a local configuration in XML files. In this way, the configuration that is performed in the Configuration Console is offline and is propagated to runtime services via a deployment function.

Such an approach provides a more manageable environment (configuration changes are enabled in batches) and an ability to restore very easily previous versions of the configuration. This also means that runtime services have fewer restrictions on physical distribution, as they do not have to read the configuration directly from a centralized location.

Management-wise, the Configuration Console supports an extensive multi-tenancy, a hierarchical tenant structure, and configurable security policies. Besides that, any change that is made in the configuration is logged at the attribute level and can be reviewed within a context at any given moment (for example, who changed priority from 10 to 20 for this process).

The Configuration Console UI, although web-based and pure HTML/JavaScript, is extended with a number of AJAX components - which enables a rich-client-like look and feel.

Management

iWD Manager is a thin-client GUI application that enables monitoring and manual management of tasks that are handled by the iWD. For more details on iWD Manager, see the *iWD Manager Guide*.

Task Life Cycle

The following sequence diagram illustrates the overall task life cycle in iWD:

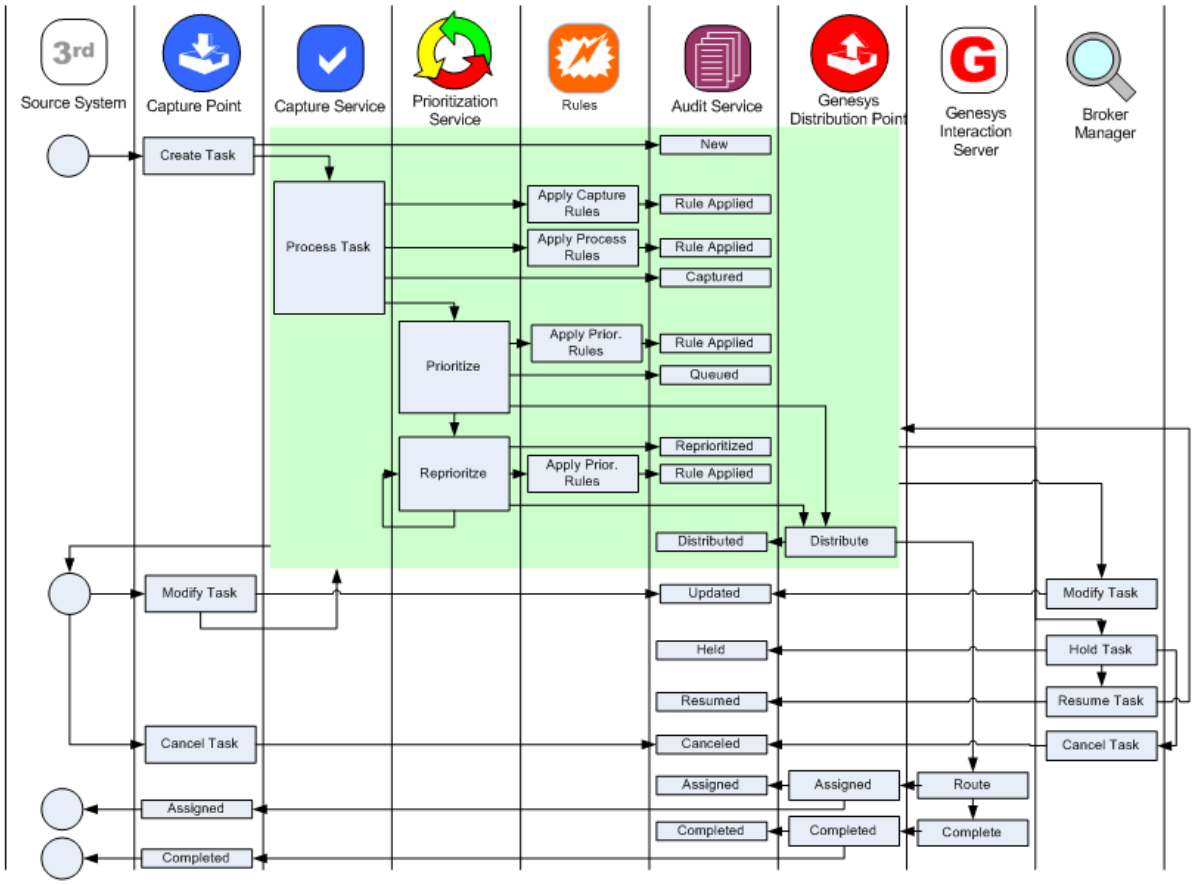

Figure 4: Task life cycle

The top row in the diagram indicates the performer of the action and grey boxes indicate action states. The exit links indicate possible state changes after the given action.