

Chapter 23

Planning a Workflow Application Deployment

This chapter provides information to help you plan an SDX deployment. The chapter describes deployment strategies for the SDX Workflow application. The strategies are divided based on the expected load of the system, integration with external OSSs, and distribution requirements.

This chapter contains the following sections:

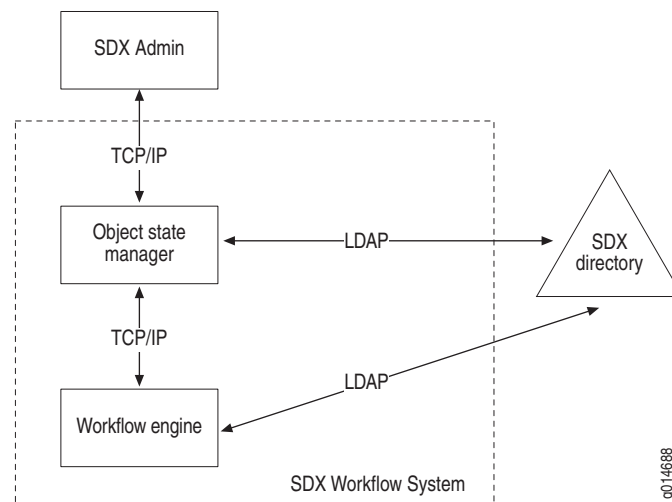
- Basic Deployment on page 422
- Centralized Deployment on page 423
- High-Availability Deployment on page 424
- Externalized Deployment on page 425
- Distributed Deployment on page 425
- Web-Based Deployment on page 427
- Mixed Deployment on page 428

Basic Deployment

A basic deployment is defined as low load, no external OSS, and a single site with no redundancy.

In this situation, a single object state manager (OSM) manages the state of all transactional objects, and a single workflow engine executes workflows. The OSM can even be located in the same host as the workflow engine. This setup is typical for integration with the DirX directory server for offline processing.

Figure 32: Basic Deployment Architecture



Because only one site hosts the Workflow application, the transactional objects, which are the directory entries that trigger workflows and whose consistency is governed by the OSM, need to be replicated and accessible in only one of the directory servers, the one used by the OSM.

Centralized Deployment

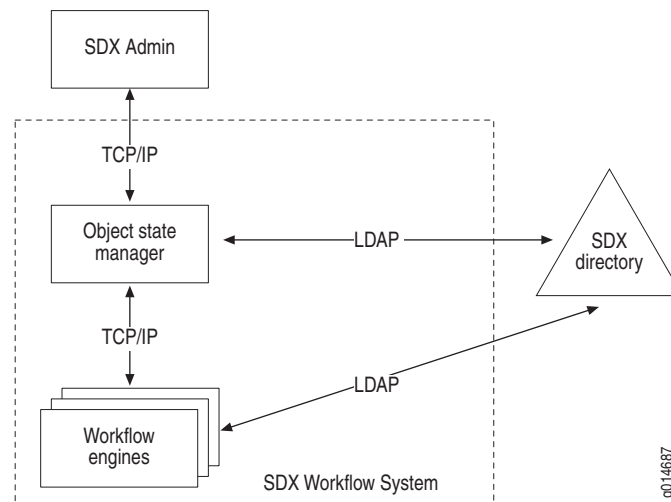
Centralized deployment is characterized by high load, no external OSS, and a single site with no redundancy.

In this situation (see Figure 33), a cluster of workstations execute workflows and manage the state of the transactional objects. To handle the high load, workflow engines are placed in several hosts. Only one OSM is necessary, because no redundancy is required. The OSM is located in one of the hosts that is allocated to one of the workflow engines. Each workflow engine is configured to report to the same OSM, and the OSM is configured to be associated with all workflow engines.

In this case, the SDX component, like the SDX Admin tool, is the client of the Workflow application.

Because only one site hosts the Workflow application, the transactional objects in the SDX directory need to be replicated and accessible in only one of the directory servers, the one used by the OSM.

Figure 33: Centralized Deployment Architecture



High-Availability Deployment

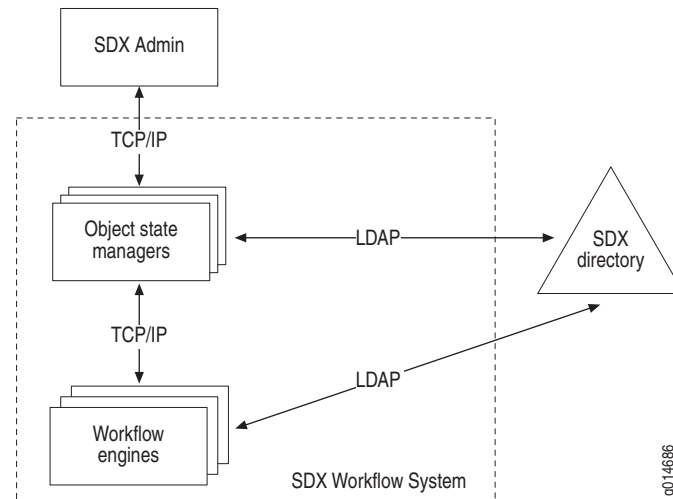
High-availability deployment is characterized by high load, no external OSS, and a single site with redundancy.

This deployment is very similar to the centralized deployment; the only difference is the use of multiple OSMs (see Figure 34 on page 424). The OSMs are replicated in different servers to improve the overall availability of the Workflow application. As a bonus, the load can be shared among them.

Like the previous case, an SDX component is the client of the OSMs. The client can issue requests to the various OSMs in a round-robin fashion. For this process to work, each OSM is associated with all workflow engines, each of which reports to a different OSM. With this setup, an engine can be reached through different OSMs, and the load is also shared. If an OSM fails, the reports destined to the OSM are queued in the engine until the OSM comes back up. If an engine fails, the requests are routed to the other OSMs.

Because only one site hosts the Workflow application, the transactional objects in the SDX directory need to be replicated and accessible in only one of the directory servers, the one used by the OSM. For increased availability, you can have more than one directory server. However, you must add directory servers with care so that the additions do not affect the performance of the OSM because of the shadowing agreements of the transactional objects. If you add a directory server, only one server should shadow the transactional objects, the one used by the OSMs. The workflow engines can use all directory servers.

Figure 34: High-Availability Deployment Architecture

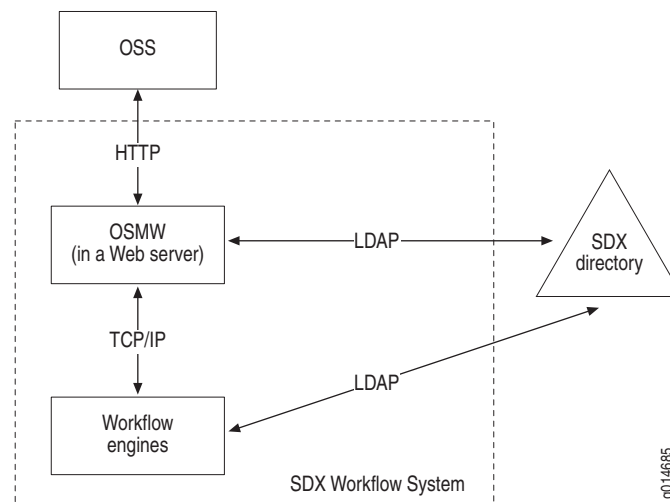


Externalized Deployment

Externalized deployment is characterized by high load, an external OSS, and a single site with no redundancy. This situation is basically the same as centralized deployment, but an external OSS controls the provisioning process (see Figure 35 on page 425). This case can be used, for example, for a customer care system that concentrates customer requests and communicates with the Workflow application to take care of the provisioning tasks.

Instead of using the regular version of the OSM, the OSMW is preferred, because it provides a friendlier interface based on XML over HTTP. The OSMW is hosted as a Web application in a Web server that is compatible with Java servlets.

Figure 35: Externalized Deployment Architecture



If availability is an issue, the Web server takes care of it, in which case the usual techniques for Web server replication can be used.

Distributed Deployment

Distributed deployment is characterized by a high load, no external OSS, and multiple sites with redundancy.

This case is equivalent to high-availability deployment, but is extended to multiple sites. A multisite deployment is necessary only if the OSS issues provisioning requests in more than one site; for example, for geographic reasons. This deployment guarantees a consistent view of the global state of all provisioning requests from every point of the service provider network that accesses the directory.

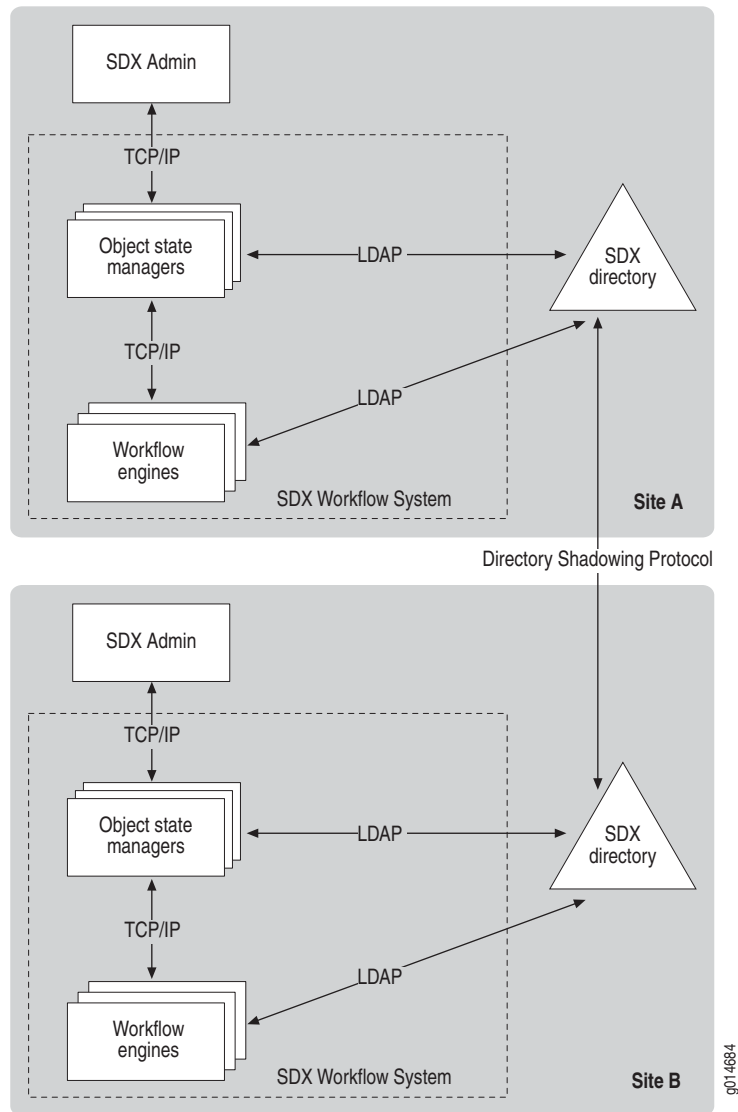
A multisite deployment of the Workflow application uses the SDX directory as the only means to communicate between sites. This means that extra configuration on the service provider administrative network is not necessary for the system components to cooperate.

Provisioning requests cannot trigger workflows across sites; that is, the OSM must be associated with workflow engines located on the same site. The workflow definitions, however, can be deployed through the directory and be accessed everywhere.

For this setup to be possible, the only requirement from the directory point of view is that the transactional objects be accessible on at least one directory server per site, because they are used to communicate between OSMs in different sites.

Figure 36 shows a typical distributed deployment.

Figure 36: Distributed Deployment Architecture



Requests issued by the SDX component on site A cause workflows to be run on site A only. The same is true for site B. However, if requests are issued on different sites for the same object—for example, the same subscription—the OSMs that handle these requests coordinate through the directory to ensure that these requests are properly executed. This coordination means that the SDX component on site B can see the state of an ongoing or completed process without needing to communicate with the other SDX component on site A.

Other sites can be easily added by providing a directory server that replicates and accesses the transactional objects, which allows gradual expansion of the system.

We do not recommend that the transactional objects be included in any shadowing agreement. Instead, the transactional objects must reside in the master directory server, and all accesses to it must be done through referrals.

Web-Based Deployment

Web-based deployment is characterized by high load, an external OSS, and multiple sites with redundancy.

Web-based deployment is the natural extension of the distributed deployment architecture in which, instead of having an SDX component as an OSS, Web-based software (for example, a portal) is used to allow customer self-care. The only difference from the distributed scenario is use of the object state manager for the Web (OSMW) instead of OSM. It can be deployed together with the Web servers that host the OSS application or on dedicated servers, according to security or other requirements.

Mixed Deployment

Mixed deployment is characterized by high load, external and internal OSSs, and a single site with redundancy.

Mixed deployment is similar to the high-availability scenario, but it also includes the externalized case (see Figure 37). This architecture solves the problem of using both external (Web-based, for example) and internal OSS applications.

The important detail here is the use of private workflow engines for the OSMW, because the external OSS application does not communicate with the OSMs that communicate with the internal OSS application.

The natural extension of this case is to distribute it on multiple sites. The idea is the same, but the distributed and Web-based cases are mixed.

Figure 37: Mixed-Deployment Architecture (Most Protocol Labels Omitted for Clarity)

