Solution Deployment Options

Solutions created with the Servoy platform can deployed to users either though a client-server architecture with a central Servoy Application Server or can be installed locally using the Servoy Runtime Client.

The majority (>98%) of all deployments use the client-server architecture.

In This Chapter

Client-server Deployment Architecture

Servoy client-server deployment architecture consists of the Servoy Application Server and the Smart and Web Client.

Besides the aforementioned Smart and Web Client, there is also the Headless and BatchProcessor Client, both of which operate headlessly on the Servoy Application Server

Servoy Application Server

The Servoy Application Server is the central hub in the client-server architecture:

- It has the connections to databases and provides access to them for the clients
- it hosts the solutions so users can launch them
- it hosts the clients that run server-side, like the Web, Headless and BatchProcessor Clients
- provides databroadcasting services to the clients
- Provides the admin interface
- etc.

Clients

Solutions can be run in one of the available clients.

Smart Client

The Smart Client is a native desktop client, that runs on the user's machine and interacts with the Servoy Application Server to retrieve and send data

The Smart Client install itself when launched for the first time and on every consecutive launch will automatically update itself if a new version is available.

Web Client

The Web Client is a browser-based client, where the User Interface of the solutions is rendered in the browser in pure HTML, CSS and JavaScript (no plugins) and all the business logic runs on the Servoy Application Server.

Headless Client

The Headless Client is, as the name suggests a headless client, meaning it does not have a User Interface. The Headless Client runs on the Servoy Application Server and can run a solution and expose both the business logic and data layer of the solution through a Java API.

As the client does not provide a UI, the client cannot be launched directly by a user, but only programatically.

BatchProcessor Client

The BatchProcessor Client is based on the Headless Client

Servoy Smart Client: A native application that runs on the users machine
Servoy Web Client: A browser based client, based on pure HTML and CSS (No browser plugins required)
Servoy Headless Client: A programmatic Java interface to the solution, to be used programatically expose and use both the business logic and datalayer defined in the solution
Servoy Batch Processor: A headless client that runs on the Servoy Application Server to perform tasks like batch processing, either periodically or when triggered by an external event.

The Servoy Application Server is available in two variations:

1. Integrated with a Tomcat application server

2. As WAR

==========

Servoy supports in basis 2 ways to deploy solutions:

1. Client/Server setup
   In this setup a central Servoy Application Server hosts the solution and users connect to the Servoy Application Server over a network (LAN/WAN/the internet) to launch one of the available Servoy Clients and run the solution
2. Local
   In this setup the so-called Servoy Runtime Client is installed on the machine of the user and the Runtime Client is launched locally by the user.

Client Server Deployment

Client Server deployment involves a centralised server which runs Servoy Server from which any of the Servoy clients can be launched. Although technically the same Client Server deployment can be implemented in three different ways:

- On Premises deployment
- SaaS deployment
- Deployment in the Cloud

On Premise Deployment

On premises deployment typically means that the application server is located on the same premises as the users of an application.

Optimal configuration:

- Dedicated Servoy server
- 100 or 1000 Mbit network connection
- no firewalls or proxies between server and clients
- 2 way socket enabled
- Java 6 or higher
- http tunnel disabled

Servoy wil run fine without the optimal configuration but with the configuration above there will be less cpu usage, less memory usage, better performance and less potential network issues

SaaS Deployment

Servoy can also be deployed using SaaS. There are a couple of ways to do this:

- on your own infrastructure (at a datacenter or not)
- in the cloud, see Deployment in the Cloud

Servoy supports the main three SaaS paradigms:

- Multi tenancy: all data is stored in a single database and the Servoy application separates the data
- Multi database: one database per customer/tenant
- ASP: One installation per client.

For SaaS deployment the server installation and management is identical to a normal installation.

Cloud Deployment

Servoy can be deployed in most cloud computing environments. Essentially all you need is a database (this can be a third party or the included PostgreSQL database) and the possibility to run a Java process (Java 5 or higher, Java 6 recommended).

Servoy Server can then be installed on the server, either with the GUI installation (if a GUI is available) or command line. It is also possible to copy a directory of Servoy from one server to another. As there are some batch scripts it is recommended to only make copies between similar operating systems. Additionally if the included PostgreSQL database is used the binaries are not cross platform.

To increase availability or performance it is possible to use Servoy Cluster and combine multiple servers.

Stand-alone Deployment

Servoy can be deployed in two ways for stand alone usage.

- Client/Server: both a Server and Client run on the same machine
- Servoy Runtime: a special stand alone version of Servoy

Client/Server

In this case a server is installed locally and a Smart or web client is deployed from that server to the local user

Runtime

Using the Servoy Runtime builder a runtime is generated that contains a doubleclickable executable. No server installation is necessary and the folder in which the runtime resides is portable (it can be moved between different machines as long as they have the same host operating system)

Pro's & Con's

- Client/Server: a local server has to be installed, with Runtime this is not necessary
- Client/Server can support multiple database connections, runtime only one local
- Client/Server can be enhanced with Mobilink to enable synchronisation
- Client/Server also allows other users to connect to the server (essentially a peer to peer scenario)