6 Multi-Tenancy

Originations multi-tenancy helps eliminate the operating cost and effort required in supporting, maintaining and upgrading IT infrastructure, subsequently freeing up significant management and resource bandwidth.

For this release, current multi-tenancy allows "multi instance" solutions, providing customers multiple dedicated instances for a single business solution. This "one size fits all" method, provides the following benefits:

- One single installation with no requirement to clone and maintain several images of the same solution, reducing resource cost.
- Adding a new tenant is a configuration change, without impacting the activity of existing tenants, which provides quick activation of new tenants.

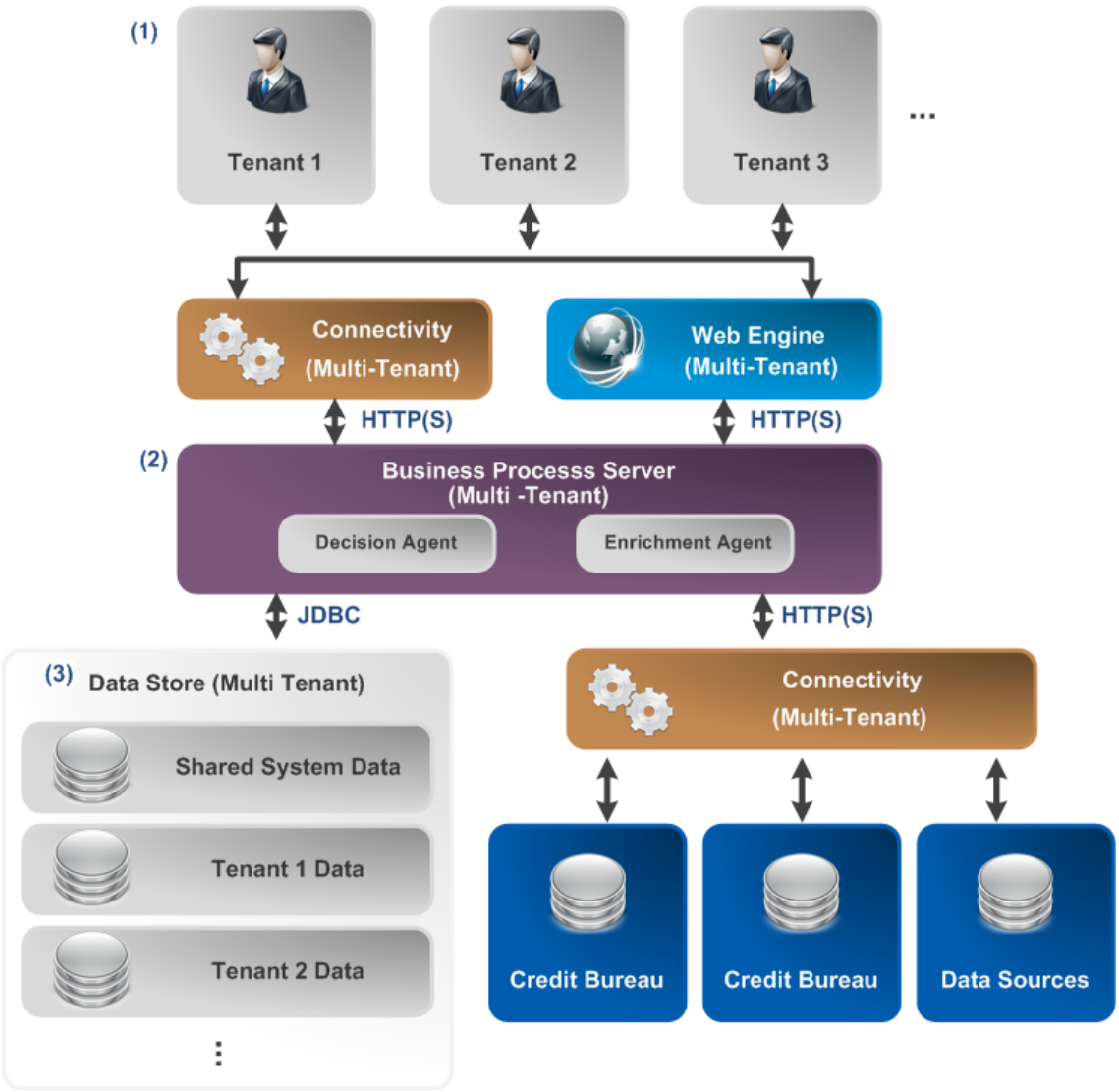
Current multi-tenancy also fits for hosted solutions, including web or screenless:

- hosted by Experian
- hosted by a customer that needs to host a service for several subsidiaries

With a multi tenant setup, information can be configured per tenant, such as:

- deploying strategies or jobs per tenant.
- configuring tactical and dynamic parameterization per tenant
- traces, error logs, and audit trails per tenant
- secured internal or external (LDAP) user administration per tenant
- customizing web application appearance look and feel per tenant (using CSS)
- business activity monitoring per tenant
- document management per tenant
- external authentication directories and security policies

Diagram below shows multi-tenancy concept in Originations:



(1) Clients are referred to as 'tenants' in this document.

(2) BPS is multi tenant and supports vertical and horizontal scalability.

(3) Tenants are managed using dedicated interface, which require no additional installation to add a new tenant. Applicant data is stored in one schema per tenant. Secured data partitioning (one database schema per tenant; for Oracle database only) ensures no risk of data corruption across tenants. The database stores tenant data in separate database tables. These tables use TENANT ID to differentiate tenant database tables. For example, database tables that display EDA_TENANT1 store TENANT1 data.