

Multitenancy scenarios and use cases

- [Multitenancy Status Quo \(October 2014\)](#)
- [Definition of terms](#)
- [Scenarios](#)
- [Use cases / requirements](#)

Multitenancy Status Quo (October 2014)

Resources and discussion on MultiTenancy in Sling from the last months:

- Current implementation of Sling Tenant: <http://svn.apache.org/repos/asf/sling/trunk/contrib/extensions/tenant> (no documentation available besides the source code currently)
- Other Wiki pages:
 - [Multitenancy Support](#) (First draft Felix Meschberger from 2009)
 - [Multitenancy Support Integration](#) (proposal from Andreas Schaefer from February 2014)
 - [Context-specific configuration use cases](#)
 - [Ideas for a multi-tenant and multi-module content model](#)
- Recent discussions in the Mailing list:
 - Proposal from Andreas Schaefer (February 2014): <http://apache-sling.73963.n3.nabble.com/Tenant-Implementation-in-Sling-td4031217.html>
 - RT from Carsten Ziegeler (August 2014): <http://apache-sling.73963.n3.nabble.com/RT-Multi-Tenancy-td4037558.html>
 - RT from Bertrand Delacretaz (August 2104): <http://apache-sling.73963.n3.nabble.com/RT-The-various-roles-of-multitenant-content-td4037819.html>
 - Proposal from Stefan Seifert (October 2014): <http://apache-sling.73963.n3.nabble.com/PROPOSAL-Context-specific-configuration-for-Apache-Sling-Multitenancy-td4041134.html>
 - RT from Bertrand Delacretaz (January 2015): <http://apache-sling.73963.n3.nabble.com/RT-Ideas-for-a-multi-tenant-and-multi-module-content-model-tt4046683.html>

Definition of terms

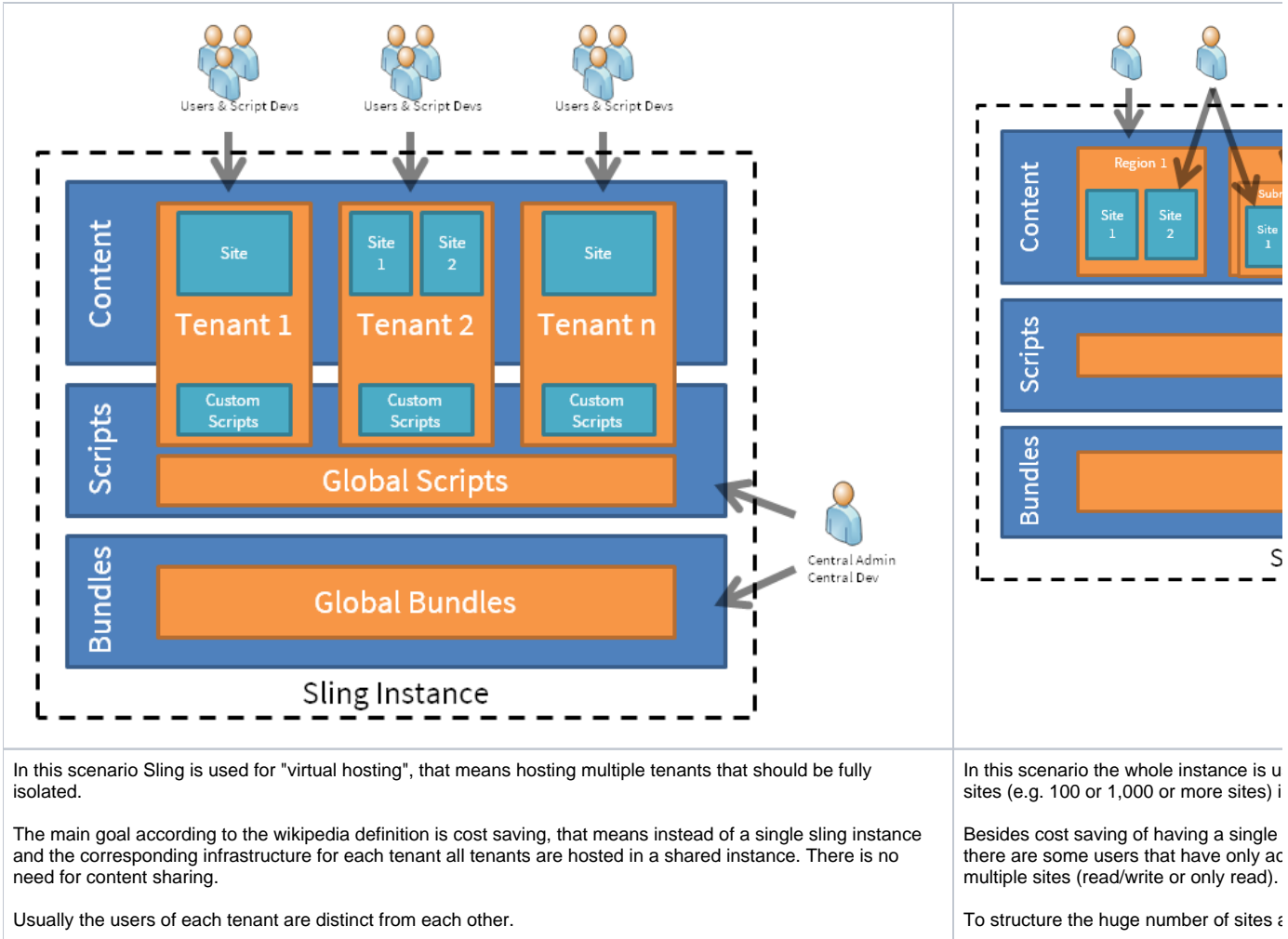
Definition of terms used in this wiki page

- **Multitenancy**
 - The generic description is found in Wikipedia: <http://en.wikipedia.org/wiki/Multitenancy>
 - The main question is: what exactly is a "Tenant" in context of Sling, and what are the requirements for customization etc. This wiki page should help to shape the answers.
- **Content**
 - Content accessible via Resources in the Resource Hierarchy (in most cases in a JCR content repository) which belongs to each tenant
 - Is stored by convention below /content/*, but this is not mandatory
 - May be associated via resource types with scripts and components globally available, or tenant-specific custom scripts
- **Configuration**
 - Configuration that customizes the system and application behavior
 - Either globally, or on a more fine-granular level (e.g. per tenant)
 - OSGi configurations are part of it, but there may be other configuration as well
- **Scripts**
 - Scripts like JSP/ESP/Sightly/Java files stored as resources in the resource hierarchy that are used by Sling to render the views of the resources
 - Can be both globally available (stored in /libs or /apps), or tenant-specific
- **Bundles**
 - Java bundles containing services deployed to the OSGi runtime
 - Usually bundles are always global, not tenant-specific. But configuration may be needed to be tenant-specific as well, which is currently only possible if the bundle implementation supports it.
- **Site**
 - A site is a collection/subtree of resources of the content that forms a logical unit used to publish a CMS-based site
 - The root of the subtree may be the homepage, with the subnodes forming the content hierarchy of the site
 - Usually there is a need for site-specific configuration
- **Region**
 - Content hierarchy is used to structure huge number of sites in separate chunks, regions can be nested to form complex hierarchies

Scenarios

After reviewing the discussions in the mailing lists there seem to be two major usecases that have the need for a Multitenancy support. Although there have some shared requirements, both have a set of special requirements with some complexity to implement that are not so relevant for the other. Of course mixes of both scenarios are possible as well.

"Virtual Hosting" Scenario	"Massive Multi Site" Scenario
----------------------------	-------------------------------



In this scenario Sling is used for "virtual hosting", that means hosting multiple tenants that should be fully isolated.

The main goal according to the wikipedia definition is cost saving, that means instead of a single sling instance and the corresponding infrastructure for each tenant all tenants are hosted in a shared instance. There is no need for content sharing.

Usually the users of each tenant are distinct from each other.

In this scenario the whole instance is used for multiple sites (e.g. 100 or 1,000 or more sites) in a single instance.

Besides cost saving of having a single instance there are some users that have only access to multiple sites (read/write or only read).

To structure the huge number of sites in a single instance

Use cases / requirements

Topic /Scenario	"Virtual Hosting" Scenario	"Massive Multi Site" Scenario
Tenant definition	<ul style="list-style-type: none"> Tenants are defined centrally and up-front Tenants are completely isolated from each other A flat list of tenants is acceptable 	<ul style="list-style-type: none"> Tenants are defined implicitly by the content structuring in regions and sites It is required to have "nested tenants" with some sort of inheritance
Content	<ul style="list-style-type: none"> A user can access only the content of his tenant, he is not able to see any content of other tenants. Inside his tenant, ACLs may prevent him from reading or writing some of the tenants content. 	<ul style="list-style-type: none"> A user can read and write all content of the sites/regions he has access to (he can have more read access than write access).
i18n	<ul style="list-style-type: none"> Tenants must be able to overlay the Internationalization so that they can provide and use their own translations 	<ul style="list-style-type: none"> It has to be possible to overlay translations per region or site. Sometimes translations to a foreign language can be shared between regions/sites, although this is limited due to different variants of the same language in different countries

Scripts	<ul style="list-style-type: none"> • Developers of the tenant can upload their own scripts with custom code to the instance. Inside the tenant only the custom scripts and the global ones can be used, no scripts of other tenants. • It is possible to overlay scripts provided globally with tenant-specific ones. It is possible to overlay both scripts at /libs and at /apps. 	<ul style="list-style-type: none"> • The global scripts provided by applications can be used in all sites and regions. Although it is possible that some custom scripts are developed specific for a region or site, it is not important to hide this extensions from other sites/regions - they may wish to use them as well. • Overlaying scripts is only used to overlay OOTB components from /libs with specific ones at /apps.
Bundles	<ul style="list-style-type: none"> • Bundles are deployed globally, all tenants can use them • The tenant has the need to adapt certain configuration settings for his tenant - this configuration is used only for his content and scripts. 	<ul style="list-style-type: none"> • Bundles are deployed globally, all regions and sites can use them • It may be required to adapt certain configuration settings per region and /or site
Security	<ul style="list-style-type: none"> • Usage of admin session in code is problematic, because it gives access to all custom scripts and all content of other tenants • All code uploaded by tenants has to be considered potentially malicious, the tenants do not trust each other • A "Tenant Admin" user concept is required 	<ul style="list-style-type: none"> • Usage of admin session is not problematic, because all code is maintained and governed centrally
Configuration	<ul style="list-style-type: none"> • It is required to be able to adapt certain configuration settings on a tenant level • It may be required to to define separate configuration for different sites of a tenant as well 	<ul style="list-style-type: none"> • It is required to support configurations on multiple levels - globally, regions, sub-regions, sites • Configuration inheritance between the levels is required • Access to the different configuration levels can be controlled by ACLs