servicenow

ServiceNow Training and Certification

Domain Separation in Service Portal

Service Portal Fundamentals

Service Portals in a Domain Separated Environment

Domain Separation allows a single instance of ServiceNow be partitioned to support multiple sub-tenants (typically customers) while retaining global administrative standards, capabilities, and reporting. Below are the types of separation:

- **Data Separation** ensures customers have visibility to view data only within their domain(s). Data Separation IS enforced in a Service Portal.
- **Process Separation** allows application configuration and business logic to be tailored within domains. Process Separation must be achieved using custom code.
- **User Interface Separation** allows forms, lists, homepages and dashboards to be tailored within domains. UI Separation must be achieved using custom Widget code and the addition of a *Domain* field on the Theme table.

While elements of the Service Portal platform such as settings, portals, pages and widgets cannot be domain-separated, the data within widgets does display based on domain. Separate portals with different URLs can be created to provide different experiences for different domains.

Instance owners must decide whether to design a single, smarter Portal that leverages parameters to dynamically incorporate design elements or to develop custom portals for each customer.

Single, Smarter Portal	
Leverage parameters in widget code **Add domain-awareness to themes**	
PROS • Less maintenance overhead • Appearance of personal touch • Data respects domain visibility automatically • Modifications to design may be data-driven	CONS • Initial implementation complexity • Limited tailoring options per domain • Custom widget code required • Changes impact all customers
Portal Per Customer	
Separate portal per customer **Custom design and branding**	
PROS • Design flexibility to meet customer needs • Leverage base instance widgets • Changes impact one customer	CONS • Additional maintenance overhead • Portal URL accessible by any user • Risk of exposing customer identity through hard-coded design • Custom code required for login behavior

Single, Smarter Portal

Parameters are typically defined as additional fields added to Company records within an instance. The company logo, primary support chat link, Knowledge Base home, and primary contacts are some common Portal elements that may differ from customer to customer rather than being statically defined on the Service Portal record. Widgets referencing these values must be customized or replaced with versions that will read the parameterized values instead. If variations from customer to customer are limited, this option may be cost-effective and maintainable.

Portal Per Customer

The creation of a Portal per customer allows for a straightforward design and implementation process, leveraging base instance Widgets. However, the overhead of maintaining multiple Portals and the risk of exposing a customer's identity via hard-coded branding may make this option less desirable.

Service Portal Tables

Portals [sp_portal]

- May not be separated due to the unique constraint on the URL suffix field
  - URL drives entry to the portal and URLs may not be overridden
  - If two customers are to access the same URL, Administrators should introduce additional parameterization logic within each Widget to make each data-driven

Themes [sp_theme]

- Sys_domain and sys_overrides may be added to the sp_theme table to allow each domain/branch to have its own Theme

Widgets [sp_widget]

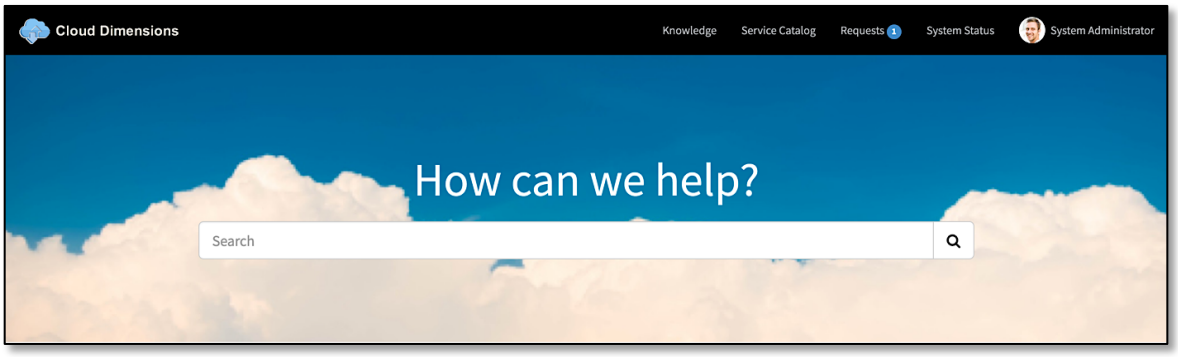
- May not be domain separated
- Portal designers have access to see all Widgets
- A Widget's Server-script may be used to retrieve dynamic customer data from the Company table

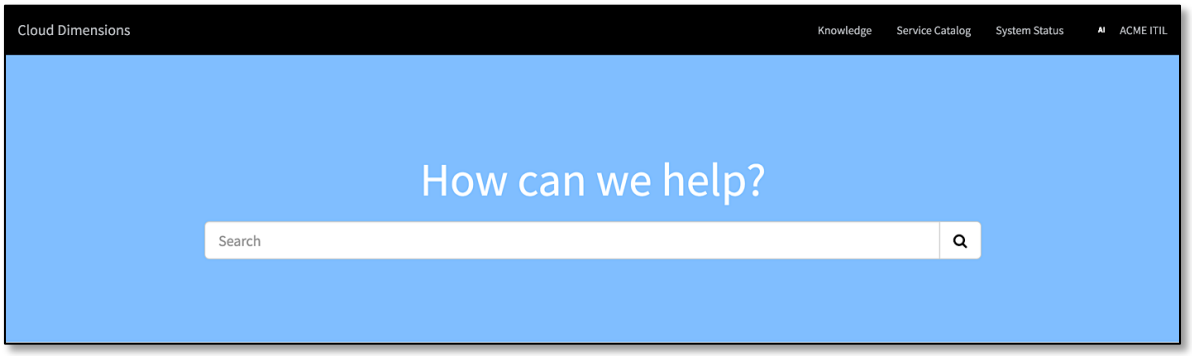
Example Code: Can be added to a Widget's Server-script to make the Portal header data-driven.

```
var cr = new GlideRecord('core_company');
cr.addQuery('sys_id', '=',gs.getUser().getCompanyID());
cr.query();

if (cr.next()) {
  data.logo = cr.banner_image;
  var a = new GlideRecord('sys_attachment');
  a.addQuery('table_name','=','ZZ_YYcore_company');   //the ZZ_YY is the prefix used
 a.addQuery('file_name', '=','banner_image');   //name of the image field

if (a.get('table_sys_id',cr.sys_id)) {  //sys_id of the record containing the Image Field
    data.logo = a.sys_id + '.iix';   //extension appended for the image itself
  }
}
```

Visual Example

- Image on the top shows a homepage developed in a Domain named *Cloud*
- Image on the bottom shows the Page when a user from a different domain tries to view it
    - Logo and background missing
    - User does not have access to those files as they are housed in another Domain





If the MSP implements a single global Portal environment, a Cloud Dimensions customer subscribed to services such as Archival, Backup, or Service Desk would see links for these Services if they are included on any Portal page. Customers not subscribed to these Services will not see them as Services are defined at the customer level which is driven from the users **Company** record.

Have the Widget Server-script check the users Company in order to determine if the content should display on the Portal page.