Multitenancy: Providing Users With a Customized Experience in a Shared Environment

Anil Gupta, Senior Sales Engineer, Embedded Solutions
This presentation may include statements that constitute “forward-looking statements” for purposes of the safe harbor provisions under the Private Securities Litigation Reform Act of 1995, including descriptions of technology and product features that are under development and estimates of future business prospects. Forward-looking statements inherently involve risks and uncertainties that could cause actual results of MicroStrategy Incorporated and its subsidiaries (collectively, the “Company”) to differ materially from the forward-looking statements.

Factors that could contribute to such differences include: the Company’s ability to meet product development goals while aligning costs with anticipated revenues; the Company’s ability to develop, market and deliver on a timely and cost-effective basis new or enhanced offerings that respond to technological change or new customer requirements; the extent and timing of market acceptance of the Company’s new offerings; continued acceptance of the Company’s other products in the marketplace; the timing of significant orders; competitive factors; general economic conditions; and other risks detailed in the Company’s Form 10-K for the year ended December 31, 2016 and other periodic reports filed with the Securities and Exchange Commission. By making these forward-looking statements, the Company undertakes no obligation to update these statements for revisions or changes after the date of this presentation.
MicroStrategy is Designed to Support Key Requirements for Embedded Applications

**Multitenancy**
How do we share resources while making sure each tenant feels like the application is their own?
How do we ensure data security?

**Integration and SSO**
How can we embed or rebrand MicroStrategy?
How do we silently sign users into MicroStrategy from our application?

**Automation**
How do we automate key processes such as user provisioning, QA, and content delivery?
Multitenancy: Providing Each Tenant a Customized Experience in a Shared Environment
Multitenancy: Not an Either/Or, But a Spectrum with Tradeoffs

- **Isolation**
  - Standalone
  - Partial Isolation

- **Sharing**
  - Partial Sharing
  - Complete Sharing
  - Hybrid

Intelligence Server

Project
Multitenancy: Not an Either/Or, But a Spectrum with Tradeoffs

- **Isolation**
  - Customizability
  - Tenant-level maintenance
  - Tenant-level resource allocation

- **Sharing**
  - Resource consumption
  - Administrative overhead
Security and Access Control for All Tenants
MicroStrategy Security supports a 3-dimensional user profile that personalizes environment, tailors reports, and protects data

BI Object Access Control (Permissions)
• Set up private workgroup folders or reports
• Impose column-level data security
  • Control development access to all objects

Feature Privileges
• Establishes customized user environment
• User sees only the functionality appropriate to their skill level

BI Security Filters
• Establishes row-level data security

Connection maps
• Segregate databases and caches
Key Capabilities for Each Architecture - Standalone

Each tenant has its own servers and metadata schema. These deployments are often on-premise or the tenants contractually require complete isolation.

### Silent Installation, Configuration, and Upgrade
- Automate the deployment and upgrade processes.
- See MicroStrategy Installation and Configuration Guide.

### Object Manager Packages, imported via Command Manager scripts
- Make it easy for an on-premise administrator or system to deploy new content.
- See MicroStrategy System Administration Guide.

### Object Name Personalization
- Use tenant-specific lingos rather than forcing a common nomenclature.
- See Supplemental Reference for System Administration.
Key Capabilities for Each Architecture – Partial Isolation

Each tenant has its own database schema and MicroStrategy project. This approach is common in cases where there are substantive differences in the tenant database schemas.

Partial Isolation

System Manager

- Iteratively deploy additional content to every project.
- See MicroStrategy System Administration Guide.
Key Capabilities for Each Architecture – Partial Sharing

Each tenant has its own database schema, but the MicroStrategy project is shared. The tenant database schemas are identical or nearly identical, but they are separate.

Connection Mapping

- Assures that each tenant’s report jobs access that tenant’s ODBC connection.
- Enable “Create caches per Database Connection” and “Create Intelligent Cubes by Database Connection” to make sure data in caches/cubes is accessed according to database connection.
- See MicroStrategy System Administration Guide.

Permissions

- Use access control lists to maintain tenant-specific folders that are invisible to other tenants.
- Assure that centrally created/deployed objects are not overwritten by tenants.
- See MicroStrategy System Administration Guide.
Key Capabilities for Each Architecture – Complete Sharing

The database schema and MicroStrategy project are shared. Every table containing tenant-specific data contains a tenant ID column

**Security Filters**

- Use security filters on tenant id attribute(s) to make sure each tenant sees only their own data.
- Add additional filtering criteria for more granular security.
- See MicroStrategy System Administration Guide.
Key Capabilities for Each Architecture – Hybrid

Some tenants share, others have some level of isolation for contractual reasons or due to substantially different databases

Some sharing + some isolation
Multitenant example: Complete sharing

Filtering of data

Tenant-specific folders

Show/Hide metrics
Personalization within a Shared Project
An application can look like it was tailor made for each tenant with little overhead

• Personalize logos
• Personalize attribute/metric names
• Personalize schema
Personalization Logos and Object Names

Use each tenant’s branding and nomenclature

Personalize logos and attribute/metric names
Personalization of Logos via HTML Tags
### Personalization of Attribute/Metric Names
Use Object Name Personalization / Internationalization Capability to Enable Tenant-Specific Nomenclature

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Tenant 1</th>
<th>Tenant 2</th>
<th>Tenant 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextAttr1</td>
<td>Active</td>
<td>Texture</td>
<td>Children’s Product</td>
</tr>
<tr>
<td>TextAttr2</td>
<td>Lifestyle Type</td>
<td>Vendor</td>
<td>(Hidden)</td>
</tr>
<tr>
<td>TextAttr3</td>
<td>Buyer</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
</tr>
<tr>
<td>TextAttr4</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
</tr>
<tr>
<td>IntAttr1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IntAttr2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IntAttr3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IntAttr4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dteAttr1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dteAttr2</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
</tr>
<tr>
<td>dteAttr3</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
</tr>
<tr>
<td>dteAttr4</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
<td>(Hidden)</td>
</tr>
</tbody>
</table>

**Note** – Object Naming and ACL’s can also be used to handle tenant-specific data requirements.
Analytics
Product demonstration

Multitenant example: Partial sharing

Filtering of data
Personalization of Logos, Object Names
Personalization of Schema
Use Object Name Personalization / Internationalization Capability to Enable Tenant-Specific Nomenclature

1) Create extra db columns of different data types (e.g., string, integer, date, double)
2) Create Attributes/Facts/Metrics on top of these columns. Keep them hidden from all tenants
3) When necessary, populate the column for a particular tenant, add a translation to the object name for that tenant’s language, and unhide the object for that tenant

Note: It is technically possible to add certain types of tenant-specific database schema customizations in a shared project, but this approach should generally be avoided.
Automation: Administer the Environment Consistently while Minimizing Effort
Automation and Administration
Powerful tools let you monitor, automate, and control your deployments with fewer administrators.
Automate Tenant Provisioning and other Processes

Command Manager Provisioning Example

Command Manager and the Web SDK support common provisioning tasks such as creating security filters, users, groups, and folders as well as assigning the security filters and modifying ACL’s.

```
CREATE SECURITY FILTER "BCBS Filter" FOLDER "\Project Objects\MD Security Filters" IN PROJECT "Hospital Billing" EXPRESSION "[Insurer]@ID=("BCBS")";

CREATE USER GROUP "BCBS Group" IN GROUP "Insurer Groups";

APPLY SECURITY FILTER "BCBS Filter" FOLDER "\Project Objects\MD Security Filters" TO GROUP "BCBS Group" FOR PROJECT "Hospital Billing";

CREATE USER "BCBS" PASSWORD "demo" FULLNAME "BCBS User" PASSWORDEXP NEVER IN GROUP "BCBS Group";

CREATE FOLDER "BCBS Reports" IN "\Public Objects\Reports\" DESCRIPTION "" HIDDEN FALSE FOR PROJECT "Hospital Billing";

REMOVE ACE FROM FOLDER "BCBS Reports" IN FOLDER "\Public Objects\REPORTS\" GROUP "Everyone" FROM PROJECT "Hospital Billing";

ADD ACE FOR FOLDER "BCBS Reports" IN FOLDER "\Public Objects\REPORTS\" GROUP "BCBS GROUP" ACCESSRIGHTS FULLCONTROL CHILDRENACCESSRIGHTS FULLCONTROL FOR PROJECT "Hospital Billing";
```
Automate Testing
Identify undesired changes in SQL, graphic, data output

```
<table>
<thead>
<tr>
<th>Seq</th>
<th>Obj. Name</th>
<th>Path (Base)</th>
<th>Modification Time (Base)</th>
<th>Status</th>
<th>SQL/DMX Data</th>
<th>Graph</th>
<th>Result</th>
<th>PDF</th>
<th>Custom Tag (Base)</th>
<th>Custom Tag (Target)</th>
<th>SQL/DMX</th>
<th>Data</th>
<th>Graph</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Average Call</td>
<td>2008-09-12</td>
<td>2008-09-12 19:19:44</td>
<td>Completed</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Customer Integrity</td>
<td>Mar 24, 2008</td>
<td>2008-08-08 15:18:48 AM</td>
<td>Completed</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Internal Audit</td>
<td>Mar 24, 2008</td>
<td>2008-08-08 15:19:48 AM</td>
<td>Completed</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Data Load</td>
<td>Mar 24, 2008</td>
<td>2008-08-08 15:18:48 AM</td>
<td>Completed</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Data Load</td>
<td>Mar 24, 2008</td>
<td>2008-08-08 15:18:48 AM</td>
<td>Completed</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Data Load</td>
<td>Mar 24, 2008</td>
<td>2008-08-08 15:18:48 AM</td>
<td>Completed</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Base
```
```
SELECT a1.REGION_ID AS REGION_ID, a1.REGION_NAME AS REGION_NAME, a1.MANAGER_ID AS MANAGER_ID, Max(a1.MANAGER_LAST_NAME), 'e1.MAN FROM CALL_CENTER_MASTER AS a1,
CALL_CENTER_ACTIVITY AS a2
WHERE (a1.CALL_CTRL_ID = 'a2.CALL_CTRL_ID') AND a1.REGION_ID > 1
GROUP BY a1.REGION_ID,
a1.REGION_NAME,
a1.MANAGER_ID,
a1.CALL_CTRL_NAME,
a1.CALL_CTRL_ID;
```

```
[analytical engine calculation steps]
1. Evaluate thresholds
2. Perform cross-tabbing
```

Target
```
SELECT a1.REGION_ID AS REGION_ID, a1.REGION_NAME AS REGION_NAME, a1.MANAGER_ID AS MANAGER_ID, Max(a1.MANAGER_LAST_NAME), 'e1.MAN FROM CALL_CENTER_MASTER AS a1,
CALL_CENTER_ACTIVITY AS a2
WHERE (a1.CALL_CTRL_ID = 'a2.CALL_CTRL_ID')
GROUP BY a1.REGION_ID,
a1.REGION_NAME,
a1.MANAGER_ID,
a1.CALL_CTRL_NAME,
a1.CALL_CTRL_ID;
```

[analytical engine calculation steps]
1. Perform cross-tabbing
```
Deploy Updates
Our change management tools allow creation of stand-alone object packages that can be applied and rolled back facilitating version control.

Distributed development environments → Create update packages → Apply update packages → Migrate to production → Undo changes if necessary

Packages
- Dashboard 1.0
- Dashboard 1.1
- Dashboard 1.2
- Dashboard 1.3

Dashboard 1.3 → Tenant A
Dashboard 1.3 → Tenant B
Dashboard 1.3 → Tenant C
Monitor Usage
Enterprise Manager enables you to understand what content is popular and which users are engaged.
Deploy Consistent, Automated Workflows
Provision a Tenant Using a Command Manager Script

CREATE SECURITY FILTER "BCBS Filter" FOLDER "Project Objects\MD Security Filters" IN PROJECT "Hospital Billing" EXPRESSION "[Insurer]@ID=("BCBS")";

CREATE USER GROUP "BCBS Group" IN GROUP "Insurer Groups";

APPLY SECURITY FILTER "BCBS Filter" FOLDER "Project Objects\MD Security Filters" TO GROUP "BCBS Group" FOR PROJECT "Hospital Billing";

CREATE USER "BCBS" PASSWORD "demo" FULLNAME "BCBS User" PASSWORDEXP NEVER IN GROUP "BCBS Group";

CREATE FOLDER "BCBS Reports" IN "Public Objects\Reports" DESCRIPTION "" HIDDEN FALSE FOR PROJECT "Hospital Billing";

REMOVE ACE FROM FOLDER "BCBS Reports" IN FOLDER "Public Objects\REPORTS" GROUP "Everyone" FROM PROJECT "Hospital Billing";

ADD ACE FOR FOLDER "BCBS Reports" IN FOLDER "Public Objects\REPORTS" GROUP "BCBS GROUP" ACCESSRIGHTS FULLCONTROL CHILDRENACCESSRIGHTS FULLCONTROL FOR PROJECT "Hospital Billing";
Summary

MicroStrategy offers rich capabilities to support multitenant embedded reporting applications.

If you find yourself doing duplicative or laborious work when developing or managing such an application, there might be a good way to automate / reduce the workload.
Thank You!