APIs and SSO
Embed Modern Analytics Into Portals

Carlos Pisonero Angulo
Senior Associate Consultant
Safe Harbor Statement

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-Q for the three months ended September 30, 2019 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.
Agenda

• Open Architecture
  • Federated Analytics
  • New REST API Families in MicroStrategy 2020

• Embedding SDK & REST API
  • Integration with Portals
  • Workflow. Combined Architecture

• Dossier vs. RSD approach
  • RSD existing functionality
  • Dossier use case

• Summary

• Q&A
Open Architecture

Federated Analytics

New REST API families in MicroStrategy 2020
Federated Analytics
The Tools You Love, The Platform You Trust

Trusted Analysis

Build Trust

Deploy Trusted Models

BUSINESS INTELLIGENCE

OFFICE PRODUCTIVITY

DATA PREPARATION

DATA CATALOG

DATA SCIENCE

EMBEDDED ANALYTICS

DELIVERED (by 2020)

ROADMAP
Federated Analytics
Extensive APIs & SDKs

- CUSTOM WEB APPLICATION
- 3rd PARTY DATA AS MICROSTRATEGY DATASET
- BUILD CUSTOM CONNECTORS
- PORTAL INTEGRATION
- WHITE LABELING
- D3 | GOOGLE CHARTS | ...
- EMBED VISUALIZATIONS IN YOUR WEB APPS
- CLOUD DEPLOYMENT

- CUSTOM WEB APPLICATION
- REST API
- PUSH API
- DATA CONNECTOR SDK
- WEB SDK
- MOBILE SDK
- VISUALIZATION SDK
- EMBEDDING API
- CLOUD API
Federated Analytics
Simplified Architecture

Data Sources → Intelligence Server → Applications Server → Web Clients

Data Sources: Databases
Intelligence Server: ODBC
Applications Server: TCP/IP
Web Clients: HTTP(S)
Federated Analytics
Simplified Architecture

Data Sources → Intelligence Server → Applications Server → Web Clients

- ODBC
- TCP/IP
- REST API
- HTTP(S)
- JavaScript
- Python
- Java
- C/C++
- JSON
- PHP
MicroStrategy REST API

Complete
Retrieve data, push data, sort data, filter data, inherits central user security

Multi Source
Data contained in different systems can be merged and accessed across any channel

Robust
Tested at more than 1000 requests / second

Well documented
## Existing REST API Families

**MicroStrategy RESTful Families**

120+ APIs available in MicroStrategy 2019

<table>
<thead>
<tr>
<th>Authentication</th>
<th>Show/Hide</th>
<th>List Operations</th>
<th>Expand Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browsing</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Cubes</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Datasets</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Datasource Management</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Dossiers and Documents</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Emails</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Library</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Misc</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Object Management</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Projects</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Reports</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>Security Roles</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>System Administration</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
<tr>
<td>User Management</td>
<td>Show/Hide</td>
<td>List Operations</td>
<td>Expand Operations</td>
</tr>
</tbody>
</table>
New REST API Families in MSTR 2020
Version-prefixed APIs

What are the implications for YOUR apps?

• Maintain application integrity after upgrades
• Common practice in API platform; developers expect this
• New APIs will be safer and faster to develop because backward-compatibility is managed via the version

When is versioning used?

• A version change is introduced when “breaking” changes are made to existing endpoints
• A “breaking” change is one that changes the format of the response, the response type, or content/functionality of the API
• Does not mean parameters changed
• Not necessarily tied to MicroStrategy version
REST API Support for Prompts in MSTR 2020

All prompts and prompt actions are supported via REST API

- Previously, only Attribute element, Object and Value prompts were fully supported via REST

- In MicroStrategy 2020, Expression and Hierarchy prompts are also supported

- Users can now view and change the prompt answers for all prompted reports via the REST API and leverage these APIs in a custom prompt UI

```plaintext
Step 1: Create report instance
Step 1a: List prompt answers
Step 2: Provide prompt answers
Step 3: Retrieve data from prompted report instance
```
Embedding SDK and REST API

- Integration with Portals
- Workflow. Combined Architecture
Embedding SDK
Embed dossiers in webpages with full interactivity, using single sign on for seamless access

- **Embed Interactive Dossiers** into applications and webpages using JavaScript

- Implement Single Sign-On so that users can **seamlessly access the dossier**, without getting prompted to log in

- Leverage the Embedding API for **navigating through pages and chapters** in the dossier, applying filter conditions, changing dossier properties, and more
Easy Integration with SSO and IDPs Using SAML
Out of the box support for leading industry solutions using SAML authentication protocol

Roles and privileges

MicroStrategy Users

SAML Authentication

Identity Providers

Sales
- Group Privileges
- Group security filters
- Group access controls

Marketing
- ✔

Group Privileges
Group security filters
Group access controls

Identity Providers

onelogin
okta
Ping
Active Directory Federation Services
LinkedIn
New in MicroStrategy 2020

Embed an instance of a dossier within a custom portal

• In MicroStrategy 2020, the Embedding SDK (embeddinglib.js) has been extended to accept the instance id of a dossier as a parameter

• What this means is a custom app can interact with a dossier via REST and then instantiate the dossier via an embedded app

User logs in as “michaelsmith”

Create an instance of a dossier

Pass “michaelsmith” as a filter to the dossier instance

Pass dossier instance ID to embedding SDK

Render manipulated dossier in web app
Federated Analytics
SSO in Embedded Apps

• How does this Combined Architecture apply to SSO in Custom Portals?
• How to get the AuthToken in SSO?
Federated Analytics
SSO in Embedded Apps

• How does this Combined Architecture apply to SSO in Custom Portals?

Data Sources → Intelligence Server → Applications Server → Web Clients

ODBC → TCP/IP → REST API

JavaScript
Python
Java
C/C++
JSON
PHP

User logs in (SAML)
Embedding SDK
Embedded Dossier shown to the user in the Custom Portal

*Cookie - iSession
Combined Architecture
Backend process – Code snippet

The **cookie** contains the information of the **Session**

```javascript
var cookie = dossier._msgRouter.iframe.contentDocument.cookie;
var split_read_cookie = cookie.split(';');

var iSession = null;
for (i=0;i<split_read_cookie.length;i++){
    var value=split_read_cookie[i];
    value=value.split('=');
    if(value[0]=='iSession'){
        iSession = value[1];
        console.log(iSession);
    }
}
```

• Then make a call to pull data from an object (report)

• And do what you need with the data

• The authentication process occurs in the backend, whereas the user logs in without being prompted in the second stage to re-authenticate
Dossier versus RSD Approach

- RSD existing functionality
- Dossier use case
RSD Existing Functionality
Set a selector with a specific value

A Report Service Document can pass specific values to a selector, so it is not rendered empty

• Functionality OOTB, excluding URL API
• Automatically update when there is no data for the current selection

- Use first
- Use last
- Number of elements

• When prompting, if the one of the above options is not selected, the user needs an extra click in the selector to retrieve one of its values, rendering the RSD empty firstly
### RSD Existing Functionality

Set a selector with a specific value

<table>
<thead>
<tr>
<th>Quarter</th>
<th>1Q 2020</th>
<th>2Q 2020</th>
<th>3Q 2020</th>
<th>4Q 2020</th>
</tr>
</thead>
</table>

No data returned for this view. This might be because the applied filter excludes all data.

No data returned for this view. This might be because the applied filter excludes all data.

No data returned for this view. This might be because the applied filter excludes all data.

No data returned for this view. This might be because the applied filter excludes all data.
Dossier Use Case
How to get the same behavior in a Dossier

An embedded Dossier <embedded SDK> can retrieve a specific value filter, provided by the REST API

Scenario

• User John has a security filter for Year = {2015, 2016, 2017 & 2018}
• John wants to see initially data for Year = {2017}
• REST API can retrieve “a” value from the filter (the first, the second)
• Embedded SDK renders the Dossier with the desired value of the filter, i.e., 2017

• User Amy has the same security filter as user John
• Amy wants to see initially data for Year = {2018}
• REST API can retrieve “a” value from the filter (the first, the second)
• Embedded SDK renders the Dossier with the desired value of the filter, i.e., 2018

In either case the Dossier is not rendered empty, thus not forcing the user to extra click the filter panel to show data
Summary
Main Ideas

• Open source, more connectivity in MSTR 2020, i.e., Paxata, Jupyter, Survey Monkey (foundation for future Transaction Services in Dossiers)

• New REST API families in MSTR 2020, version-prefixed. Also, all prompts and prompt actions supported in MSTR 2020

• Extended Embedded SDK to accept the instance id of a dossier as a parameter

• Combined Architecture Embedded SDK & REST API. Embedded Dossiers via Single Sign On in Custom Portals

• Embedded dossiers with changes on-the-fly for filter passing values, at the object/user level. No need to reauthenticate
Questions?
Thank you!

Carlos Pisonero Angulo
Senior Associate Consultant
cpisonero@microstrategy.com