

Auto-provision embedded report delivery for your customers



SOARING EAGLE
DATA SOLUTIONS
& Lucient

Tom Huguelet

 thuguelet@lucient.com

 @tomhug

 www.linkedin.com/in/tomhuguelet/



25+ years in Microsoft Database, BI, Applications



Goals for this session

- Look at the technology implementation from the business case perspective
- Examine certain design choices and understand how they support the client's needs
- Intro to the basics of embedding
(lots of supplemental resources linked at the end of the deck)
- Look at PowerShell and REST API calls that support auto-provisioning
- Enumerate the 'gotchas' that the team encountered (so far 😊)

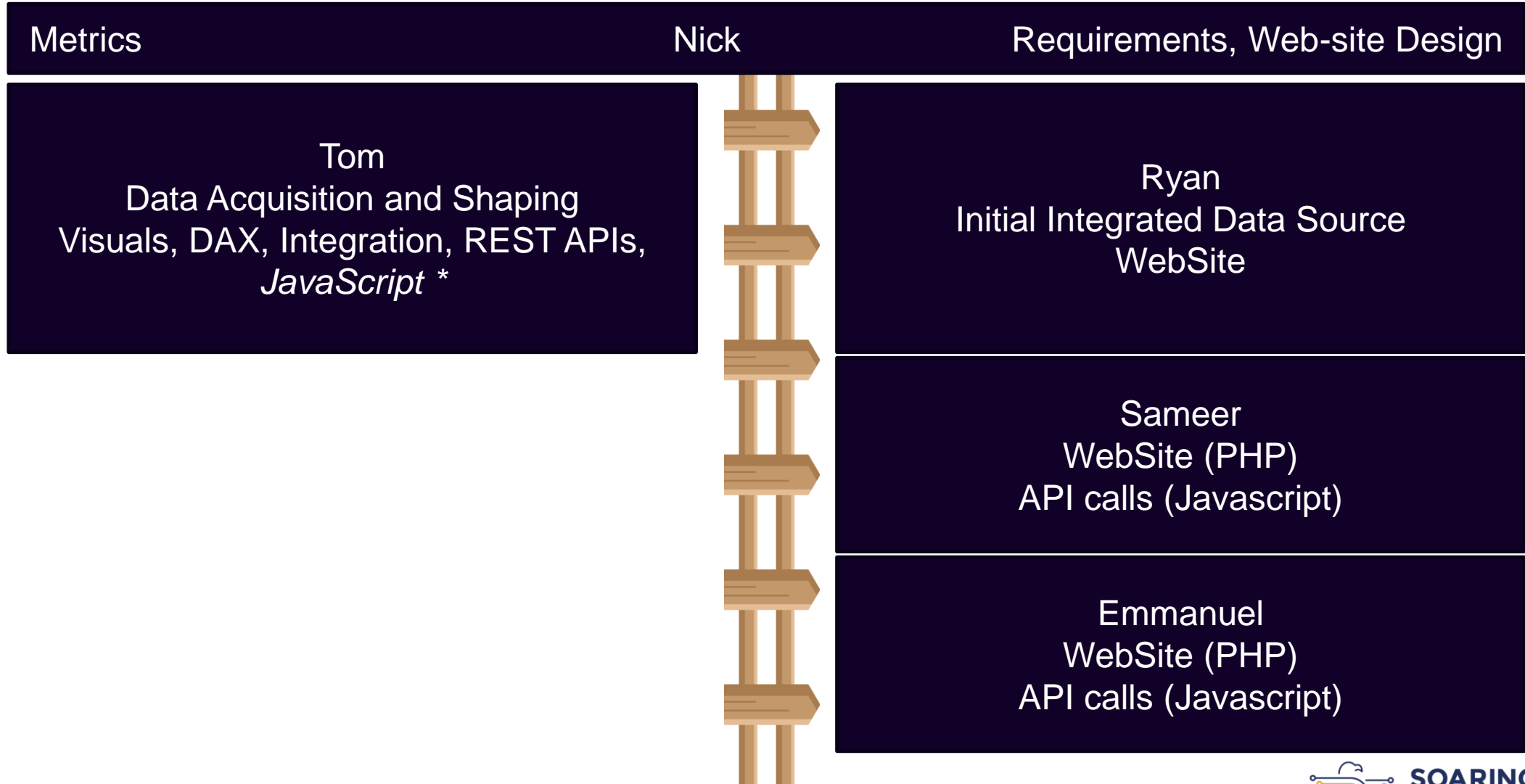
Business Case

- Amazon Resellers
- ArgoMetrix
- <https://argometrix.com/seller-intelligence-platform-sip/>
- Speed to Market a factor
- Scale to ???
 - *There were ~ 1.5 million active sellers in 2021*
 - *1,824 new Amazon sellers each day.*

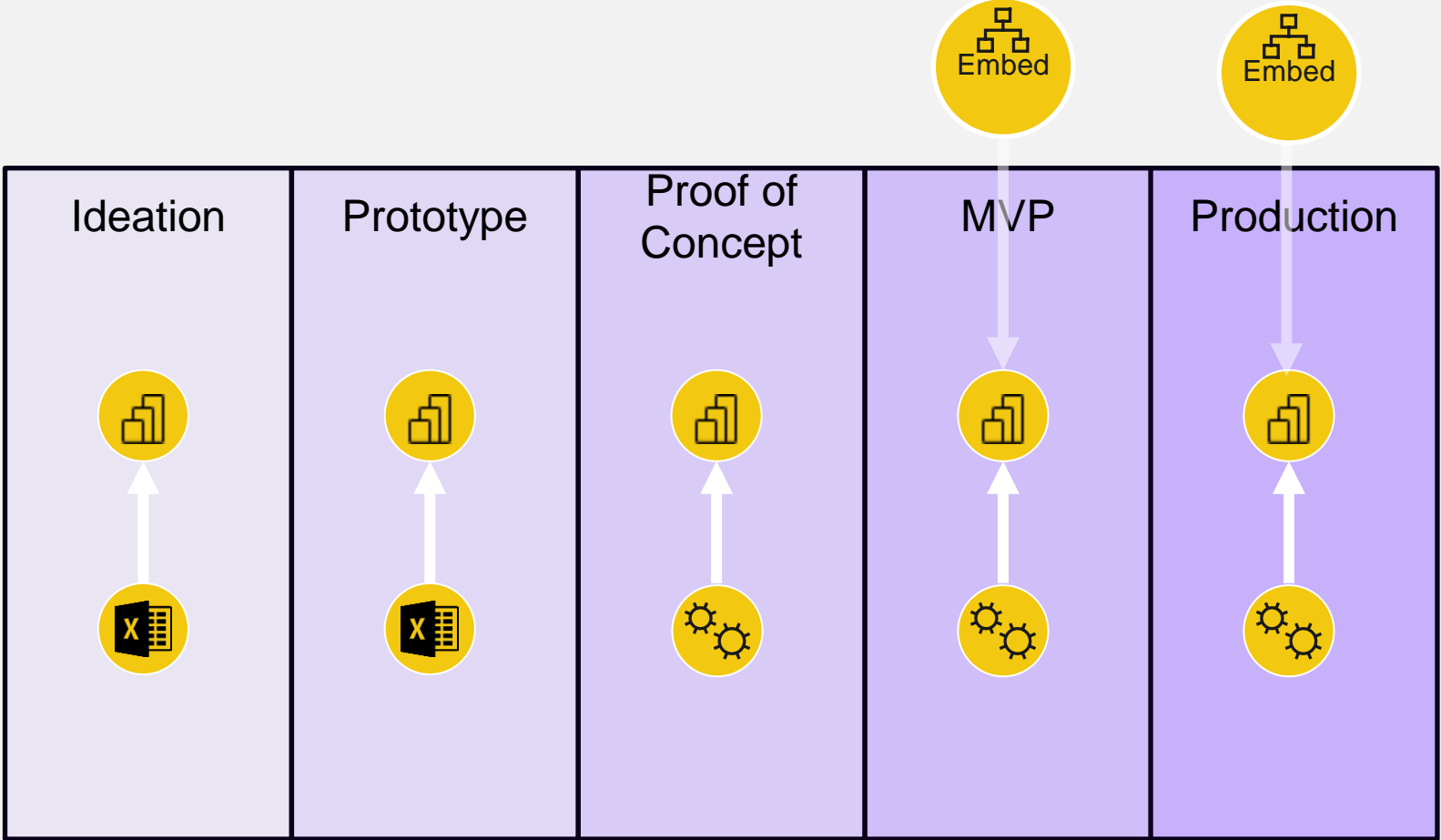
Project Team

- Blended Team
- Nick
 - CEO
 - Proxy Stakeholder i.e. Representing Amazon Sellers
- Tom
 - Architect
 - Power BI, Azure, PowerShell, REST APIs (Some JavaScript assistance)
- Ryan
 - Project Management, Back-end Design
- Sameer
 - PHP, Javascript
- Emmanuel
 - PHP, Javascript, mySQL

Project Team



Initial Prototype Development



Security Model

Seller 01

- Workspace 01
 - Dataset 01
 - Report Collection 01

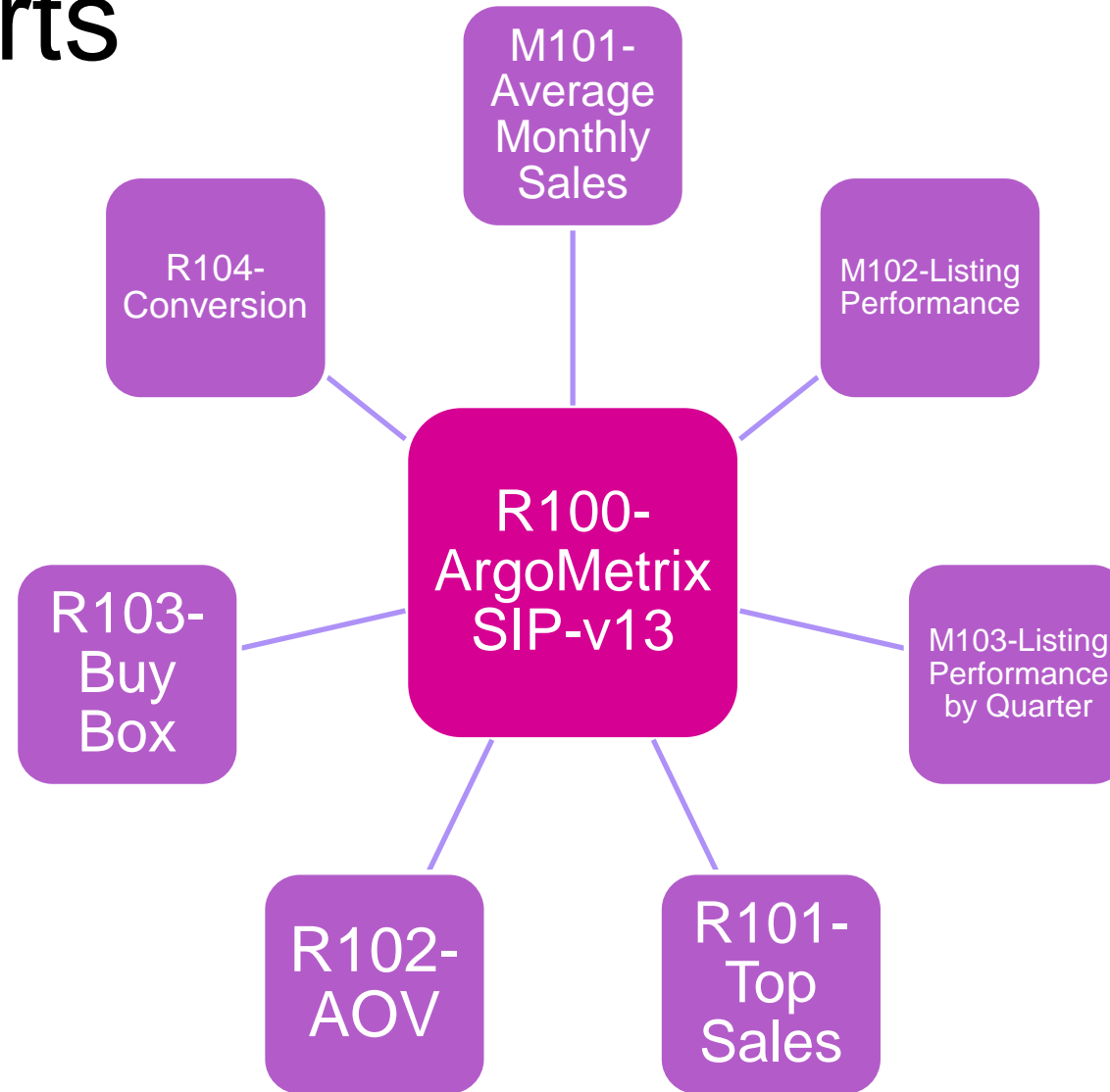
Seller 02

- Workspace 02
 - Dataset 02
 - Report Collection 02

Seller 03

- Workspace 03
 - Dataset 03
 - Report Collection 03

Thin Reports



PBIX
Files (15)

Central
DataSet

DAX Date Dimension

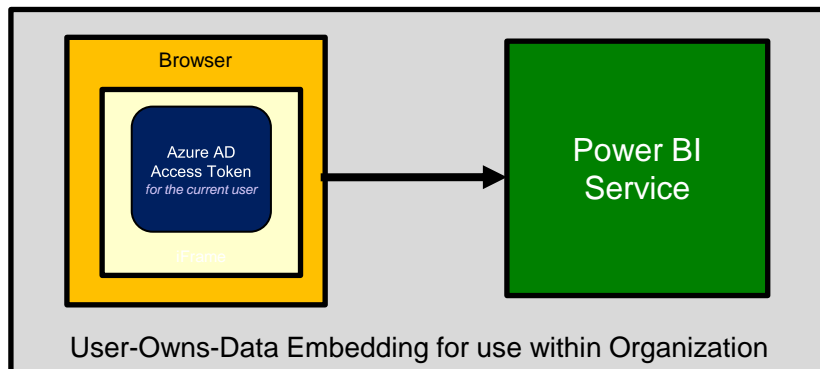
```
1 Date =
2 VAR MinYear = YEAR ( MIN ( 'Sales'[Start date] ) )
3 VAR MaxYear = YEAR ( MAX ( 'Sales'[End date] ) )
4 RETURN
5 ADDCOLUMNS (
6     FILTER (
7         CALENDAR(AUTO(),
8             AND ( YEAR ( [Date] ) >= MinYear, YEAR ( [Date] ) <= MaxYear )
9         ),
10    "Day Of Week", FORMAT( [Date], "w" ),
11    "Day Of Week Name", FORMAT([Date],"ddd"),
12    "Week Number", WEEKNUM([Date]),
13    "Month Name", FORMAT ( [Date], "mmm" ),
14    "Month Of Year", MONTH ( [Date] ),
15    "Month Year", FORMAT([Date],"MM-YYYY"),
16    "Month Year Key", FORMAT([Date],"YYYYMM"),
17    "Quarter Name", FORMAT([Date],"Q"),
18    "Quarter Of Year", FORMAT([Date],"Q"),
19    "Quarter Year", FORMAT([Date],"Q") & " " & YEAR ( [Date] ),
20    "Quarter Year Key", FORMAT([Date],"YYYYq"),
21    "Week Ending", [Date] - MOD([Date]-1, 7) + 6 ,
22    "Year", YEAR ( [Date] ),
23    "Current Month", IF( AND ( Year([Date]) = Year(NOW()) , Month([Date]) = Month(NOW()) ), "Current Month", IF ( [Date] > NOW() , "Future Month" , "Historical Month" ) ),
24    "Current Year", IF( Year([Date]) = Year(NOW()), "Current Year", IF ( [Date] > NOW() , "Future Year" , "Historical Year" ) ),
25    "Day of Month", Day([Date]),
26    "Start of Month", FORMAT([Date] - Day([Date])+1,"MM/dd/YYYY"),
27    "Current Day", If(FORMAT([Date],"YYYYMMDD") = FORMAT(NOW(),"YYYYMMDD"),"Current Day", If(FORMAT([Date],"YYYYMMDD") < FORMAT(NOW(),"YYYYMMDD"), "Historical Day", "Future Day")),
28    "End of Month", FORMAT(EOMONTH([Date],0),"MM/dd/YYYY"),
29    "Days Remaining in Month", Day(EOMONTH([Date],0)) - Day([Date]),
30    "Days Remaining in Current Month", If(FORMAT([Date],"YYYYMMDD") = FORMAT(NOW(),"YYYYMMDD"), Day(EOMONTH([Date],0)) - Day([Date]), BLANK()),
31    "Days Elapsed in Current Month", If(FORMAT([Date],"YYYYMMDD") = FORMAT(NOW(),"YYYYMMDD"), Day([Date]), BLANK()),
32    "Yesterday", If(FORMAT([Date],"yyyymmdd") = FORMAT(NOW()-1,"yyyymmdd"),"Yesterday", BLANK()),
33    "Last Week", IF( AND ( Year([Date]) = Year(NOW()) , WEEKNUM([Date]) = WEEKNUM(NOW())-1 ), "Last Week", BLANK()),
34    "Last Month", IF(EOMONTH([Date],0) = EOMONTH(NOW(),-1) , "Last Month", BLANK()),
35    "Last Year", IF(Year([Date]) = Year(NOW())-1, "Last Year", BLANK()),
36    "Days in Month", FORMAT(EOMONTH([Date],0),"dd")
37 )
```

Testing Embedded in a Sandbox Environment (Authentication)

User-Owns-Data Embedding vs App-Owns-Data Embedding

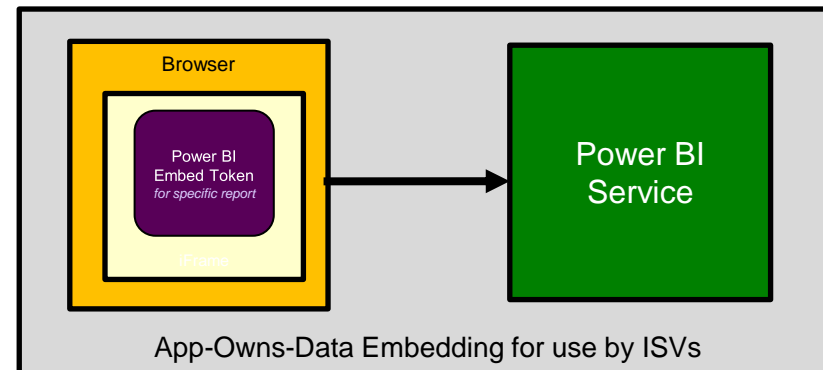
- **User-Owns-Data Embedding**

- All users require a Power BI license
- Useful in corporate environments
- App authenticates as current user
- Your code runs with user's permissions
- User's access token passed to browser



- **App-Owns-Data Embedding**

- No consumers require Power BI license
- Useful for commercial applications
- App authenticates with app-only identity
- Your code runs with admin permissions
- Embed token passed to browser



- ***No access to interactive powerbi portal***

Licensing Options

Scenario	Azure	Office
	(A SKU)	(P and EM SKUs)
Embed for your customers (app owns data)	✓	✓
Embed for your organization (user owns data)	✗	✓
Microsoft 365 apps (formerly known as Office 365 apps) <ul style="list-style-type: none">• Embed in Teams• Embed in SharePoint	✗	✓
Secure URL embedding (embed from Power BI service)	✗	✓

Enabling User-Customized Experiences

Using REST APIs for automatic Provisioning

• Lessons learned and challenges encountered

- Service Principals can't do everything needed to provision via REST API's (in our case)
 - Tip: Use the same security model from inception to deployment (duh)
- DataSet Ownership – Reports and Dataset should be owned by same account for API manipulation
- Thin Reports require extra steps to manage (i.e. redirecting to new DataSet and/or Workspace)
- Don't let your CPU Consumption exceed 65-70% capacity. If you do there will be a blocked period until garbage cleanup.
- Some objects and capabilities are only available via the (javascript) libraries – there is not 100% parity via PowerShell and REST APIS
- These are somewhat moving targets – new and better APIs are being introduced
- It is possible to have many items with the same name – GUIDs are what count
- The Power BI Portal doesn't show you everything, even if you're a Power BI Admin. You need to have a defined role in the workspace to see items via the Portal
 - Running [Get-PowerBIWorkspace](#) as the Service Principal account will allow you to see everything