

Qlik Sense Governance

Qlik Sense has very good out-of-the-box monitoring tools for the server environment. But sometimes other questions arise when I'd like to have a good insight about applications available in my Qlik Sense Site. Questions like:

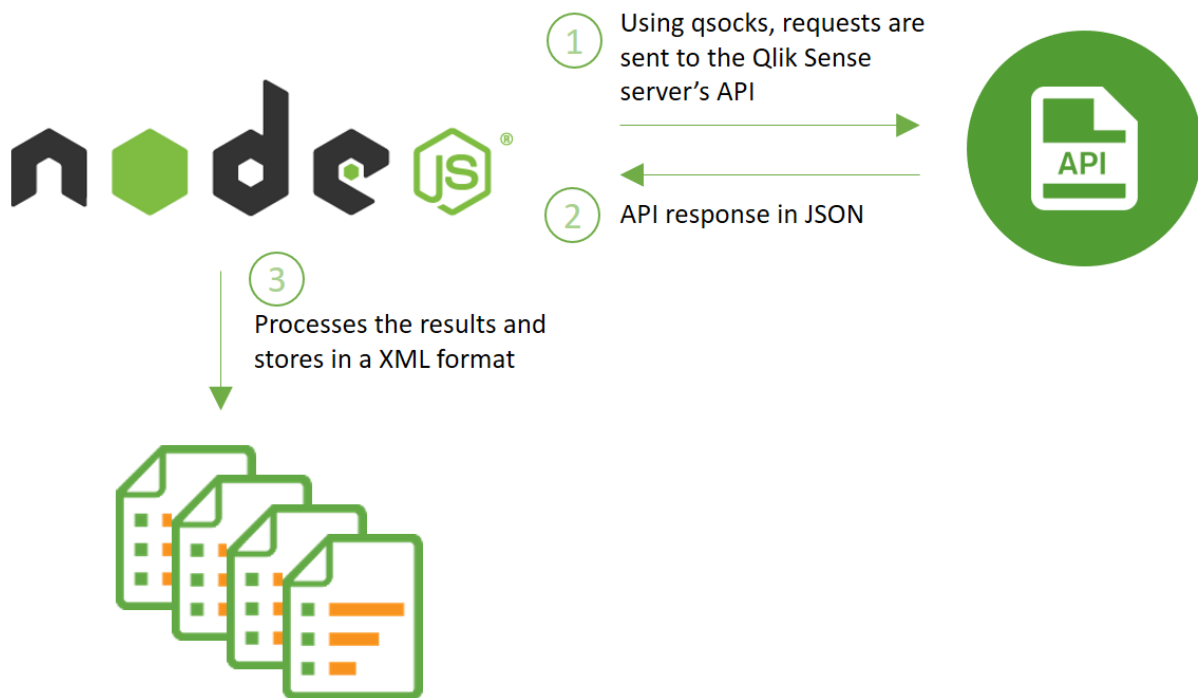
- This source is used in how many applications?
- This table/field is used in how many applications?
- This dimension or measure is in how many charts?

I am starting a new project to try to answer to this.

So far, using NodeJs, the API and fiddling around I am starting to get somewhere. Let me share with you what is happening so far.

Step 1: Node.js and Engine API

Using Node.js and some modules, of which [qsocks](#) has a very important role, I am sending some requests to the API. As a result, XML files are stored with the ("so far defined as") relevant data regarding the Qlik Sense applications available in the environment.



A quick list of the Node.js files and the expected outcome:

- **apps-info.js** – project executable with a quick helper. Defines and loads the files in the 'lib' folder in sequence.
- **lib/ app-list.js** – gets the list of applications available in the server. Stores the DocumentsList.xml file.

- **lib/app-connections.js** – gets the list of connections available to the user. Stores the DocumentsConnections.xml file.
- **lib/app-tables.js** – gets the tables of the datamodel for each one of the applications. Stores a <application-id>_KeyTables.xml file.
- **lib/app-library-dimensions.js** – gets the library dimensions for each one of the applications. Stores a <application-id>_LibraryDimensions_<dimension-id>.xml file.
- **lib/app-library-measures.js** – gets the library measures for each of the applications. Stores a <application-id>_LibraryMeasures_<measure-id>.xml file.
- **lib/app-library-masterobjects.js** – gets the library master objects for each of the applications. Stores a <application-id>_LibraryMasterObjects_<master-object-id>.xml file.
- **lib/app-bookmarks.js** – gets the bookmarks for each of the applications. Stores a <application-id>_Bookmarks_<bookmark-id>.xml file.
- **lib/app-sheets.js** – gets the sheets and its objects for each of the applications. Stores:
 - a <application-id>_Sheet_<sheet-id>.xml file.
 - a <application-id>_SheetObject_<sheet-id>_<sheet-object-id>.xml file.
- **lib/app-stories.js** – gets the stories and its contents for each of the applications. Stores:
 - a <application-id>_Story_<story-id>.xml file.
 - a <application-id>_StorySlide_<story-id>_<slide-id>.xml file.
 - a <application-id>_StorySlideItems_<story-id>_<slide-id>.xml file. (represents all the items of a specific slide except the snapshots)
 - a <application-id>_StorySnapshot_<story-id>_<slide-id>_<snapshot-id>.xml file.

```
Command Prompt

C:\Nodejs-projects\Apps-Governance>node apps-info.js -h

*****
Welcome to the Applications Information Loader
for your Qlik Server Environment
*****
Configuration instructions

This tool requires the following information.

* Qlik Sense Server
-a: Qlik Sense Server address.
   Default if unspecified: 'localhost'

-c: Qlik Sense Server certificate location (at the
    moment it was only tested with blank password
    certificates). If the path has spaces, indicate it
    using double quotes.
    Default if unspecified: file 'client.pfx' at root
    folder of this tool

* Root Admin
-ud: User Directory of the Qlik Sense Server Root Admin
    to call the server API functions
    This is mandatory.

-un: User Name of the Qlik Sense Server Root Admin to
    call the server API functions
    This is mandatory.

* Origin address for the request (this computer)
-o: This is optional. Default value is 'localhost'.
    The origin specified needs an entry in the Whitelist
    for the virtual proxy to allow websocket communication.

* Additional Options
-s: Single application mode.
    Load information of only one application given it's ID.
    Applications list and connections are still fully loaded.

-h: Launches this helper.

C:\Nodejs-projects\Apps-Governance>
```

Try it yourself!

[You can download the Node.js part of the project here.](#)

If you don't have Node.js installed, you can get it from the [official site](#). I am currently using a 5.x stable version.

Some considerations:

This is an early development. I tried to make it as stable as possible, but I am sure some issues might occur and, of course, it is open for improvements. Just let me know if you find something and we will add it.

You will need to export your server's certificate in the QMC and have it available to this project.

There is a '*Read me - Configs Walkthroug.txt*' file that should be the first thing for you to open. It explains the dependencies of packages that should be installed in your node environment.

I have only tried executing this on the same machine as the Qlik Sense Server. If you venture to try to "execute remotely" let me know how it went!

Very important: the requests sent to the Engine API will require that the applications are open and loaded to memory. I was not able to find a direct "dump the application from memory when finished loading all its information". If all your applications open at the same time consume more memory than available in your server, well... you know the rest. This is absolutely something to improve in the future.

Depending on the complexity of your objects, you might have an impression that hangs can occur. Confirm with task manager if indeed the CPU is working or not.

You'll know everything went well when you get this message:

```
Checkpoint: Applications Stories and Snapshots are loaded
Success: Finished loading all the data.
C:\Node.js-projects\Apps-Governance>
```

Step 2: Converting data retrieved from Node.js to usable QVDs

As a second step, there is a Qlik Sense application that will convert all the XML files to a QVD files structure. The idea here is that this QVD files structure can be used as the base of exploration for the environment.

The 'Sense Governance QVD Generator' application is essentially a Loading Script that crunches the XML files, storing what I thought strictly necessary. The result will be as depicted below

<input type="checkbox"/> Stories Snapshots.qvd	<input type="checkbox"/> Library Visualizations - Measures.qvd
<input type="checkbox"/> Sheet Objects - Measures.qvd	<input type="checkbox"/> Library Measures Object Usage.qvd
<input type="checkbox"/> Stories Slide Items.qvd	<input type="checkbox"/> Library Visualizations.qvd
<input type="checkbox"/> Stories Slides Link.qvd	<input type="checkbox"/> Library Measures.qvd
<input type="checkbox"/> Stories.qvd	<input type="checkbox"/> Library Dimensions Fields.qvd
<input type="checkbox"/> Sheet Objects - Dimensions.qvd	<input type="checkbox"/> Library Dimensions Object Usage.qvd
<input type="checkbox"/> Sheet Objects.qvd	<input type="checkbox"/> Library Dimensions.qvd
<input type="checkbox"/> Sheets.qvd	<input type="checkbox"/> Fields.qvd
<input type="checkbox"/> Bookmarks Fields.qvd	<input type="checkbox"/> Tables.qvd
<input type="checkbox"/> Bookmarks.qvd	<input type="checkbox"/> TablesLinks.qvd
<input type="checkbox"/> Library Visualizations - Dimensions Tags.qvd	<input type="checkbox"/> Applications.qvd
<input type="checkbox"/> Library Visualizations - Tags.qvd	<input type="checkbox"/> Connections.qvd
<input type="checkbox"/> Library Visualizations - Dimensions.qvd	<input type="checkbox"/> FieldDataTypes.qvd

Try it yourself!

[Download here the Sense Governance QVD Generator.qvf](#) file and convert your XML files to more manageable QVDs.

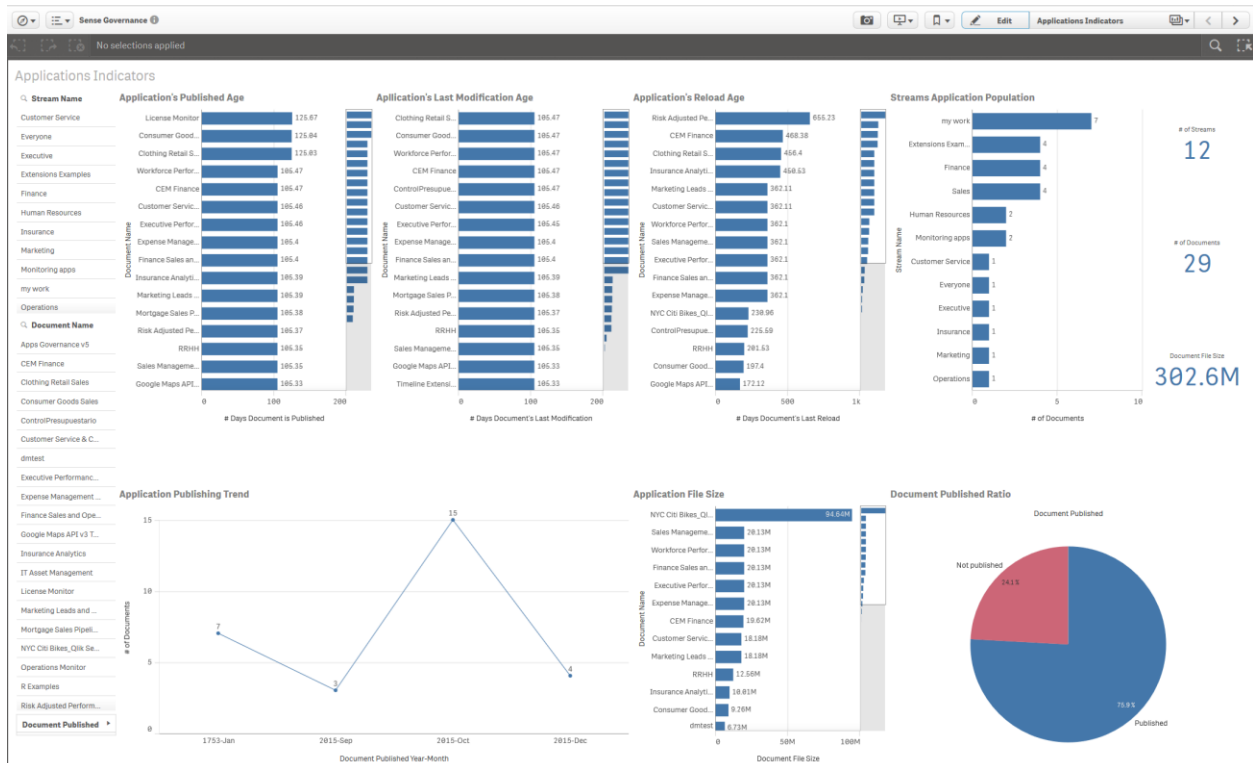
This is pretty straightforward to use. Just make sure to have the following 2 Data Connections:

- AppStructures – this is the “source” where the XML files are stored. It refers to the “AppStructures” folder that is within the Node.js project you executed before.
- GovernanceQVDs – this is the place where the QVD files are stored. You can store it in any path of your choosing.

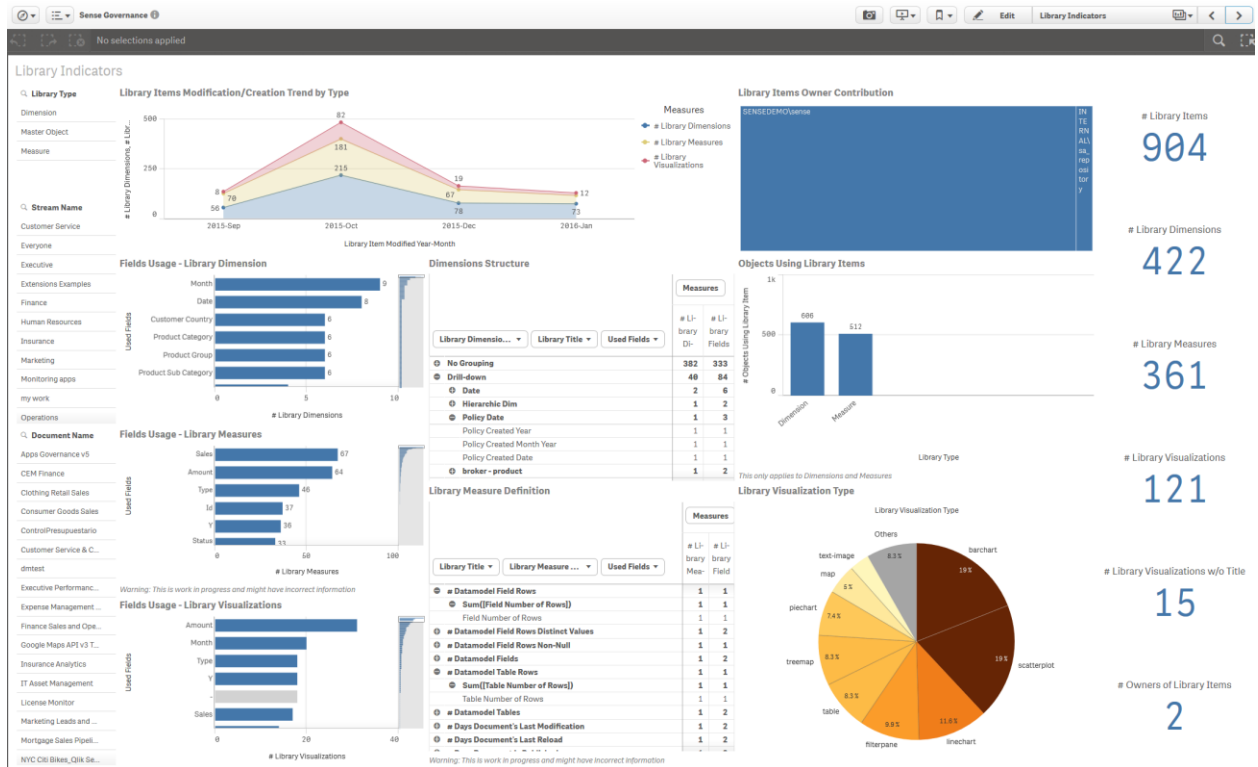
A suggestion of applications Governance base

Based on the result QVDs, and trying to find ways to answer questions customers have made to me in the last times, I created a draft of a datamodel that is focused in trying to answer the following ones:

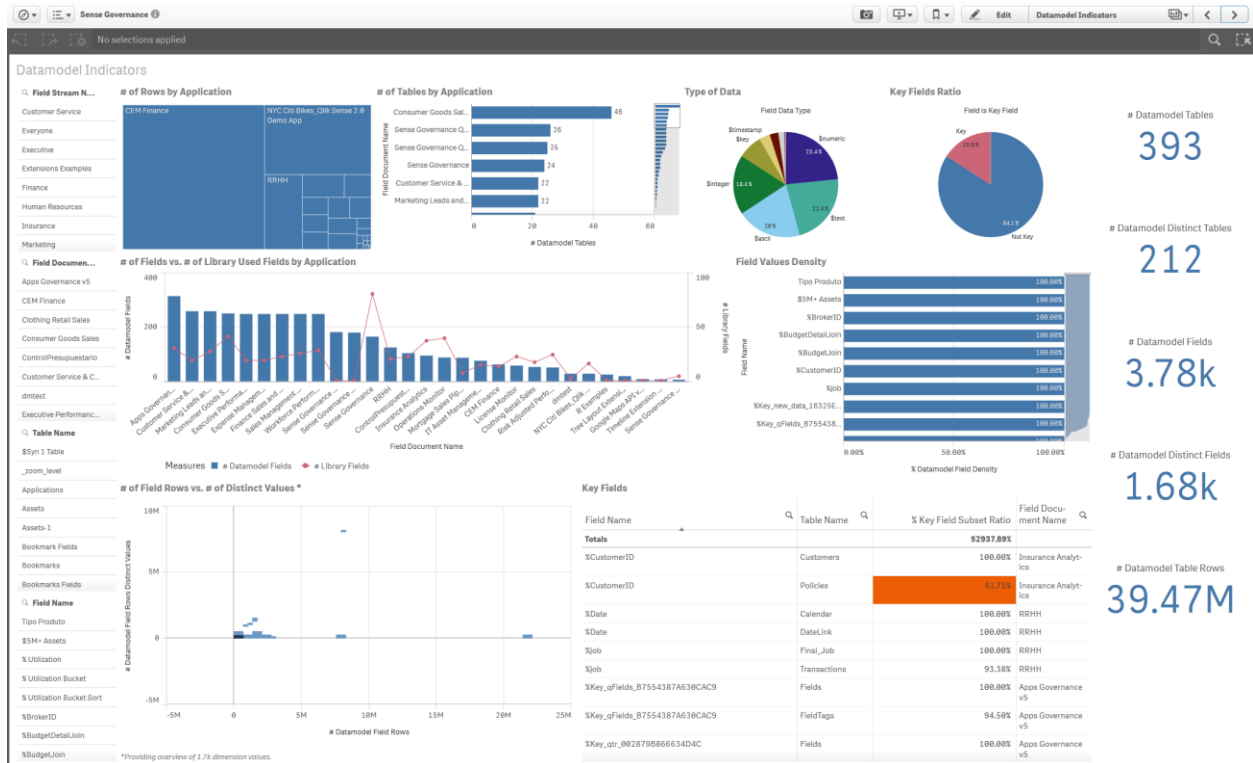
- Applications – Since when is it published? When was it last modified? How long from the last reload? What are the more populated streams? What is the publishing trend? What is the file size of my biggest apps?



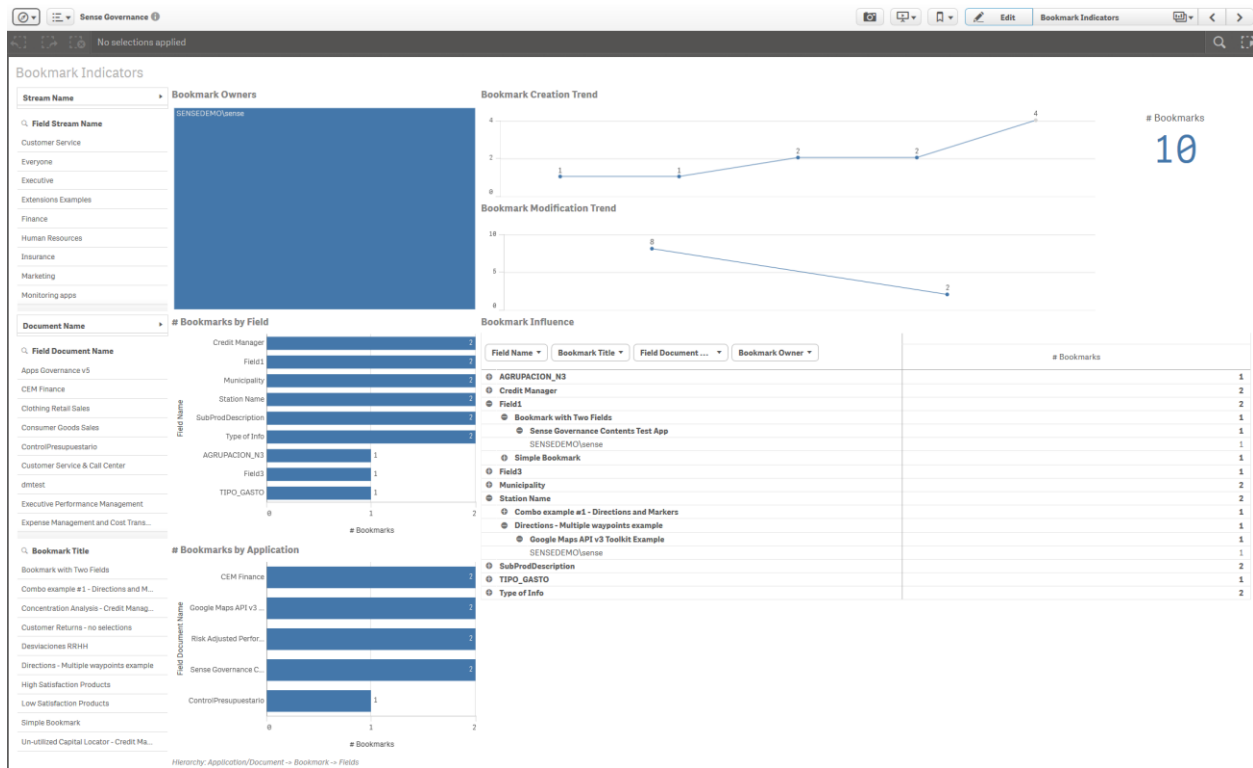
- Library Items – How many and of what kind (dimension, measures and visualizations) do I have? How are these items defined and what fields are being used in these library items? In how many objects are these library items used? What are the most used type of visualization? Who are the users that create more Library items?



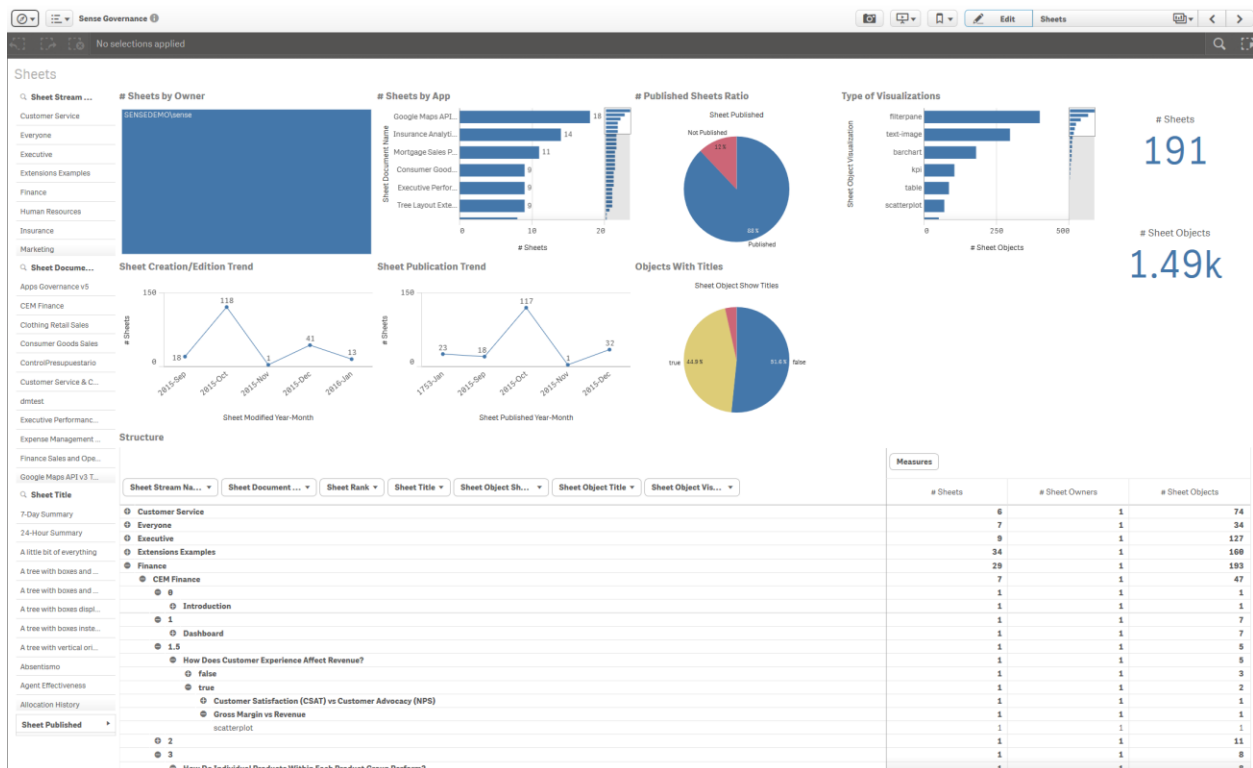
- **Datamodel** – What tables and fields exist in my applications? What applications have the greater number of rows? What type of data do I have in my fields? How many of them are key fields? And their subset ratio? How many fields are actually used in library items? Can I see a comparison between number of rows vs. number of distinct values?



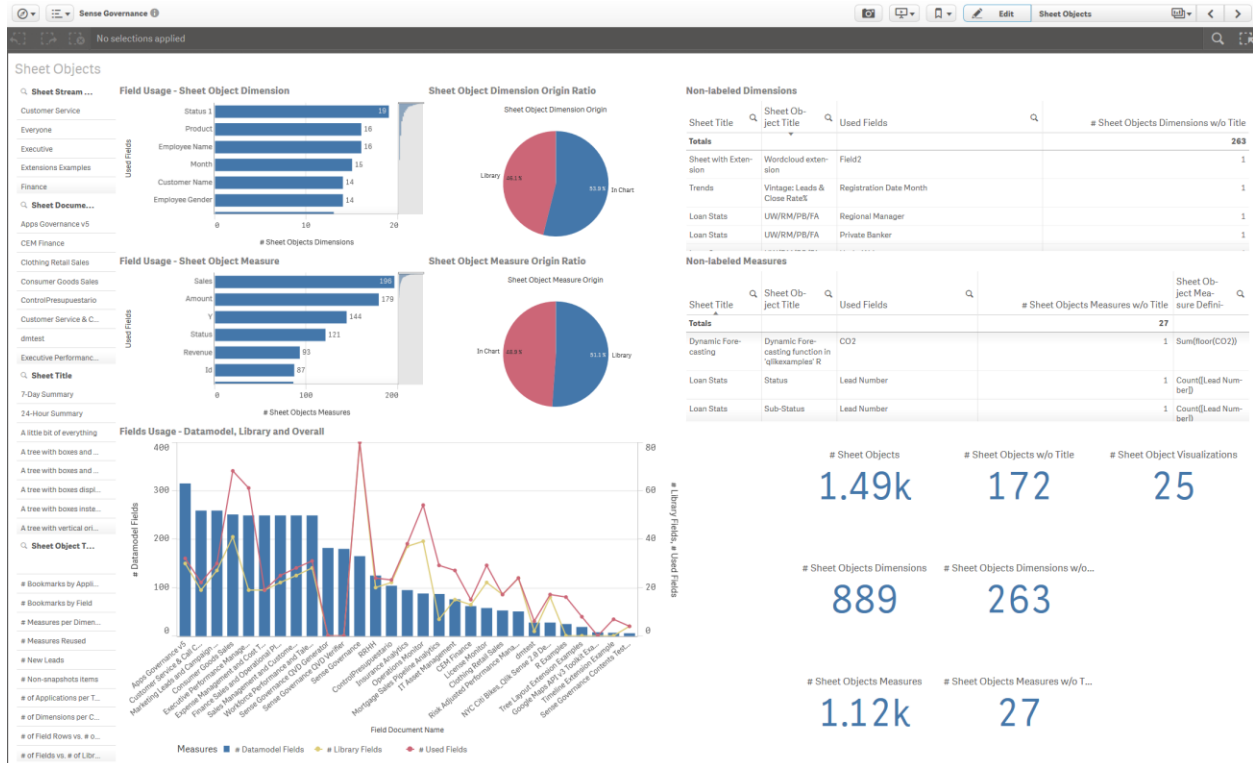
- Bookmarks – what bookmarks do I have? And how many per application? Which fields are being used in the bookmarks? Who is the bookmark owner? What is the bookmarks creation and modification trend?



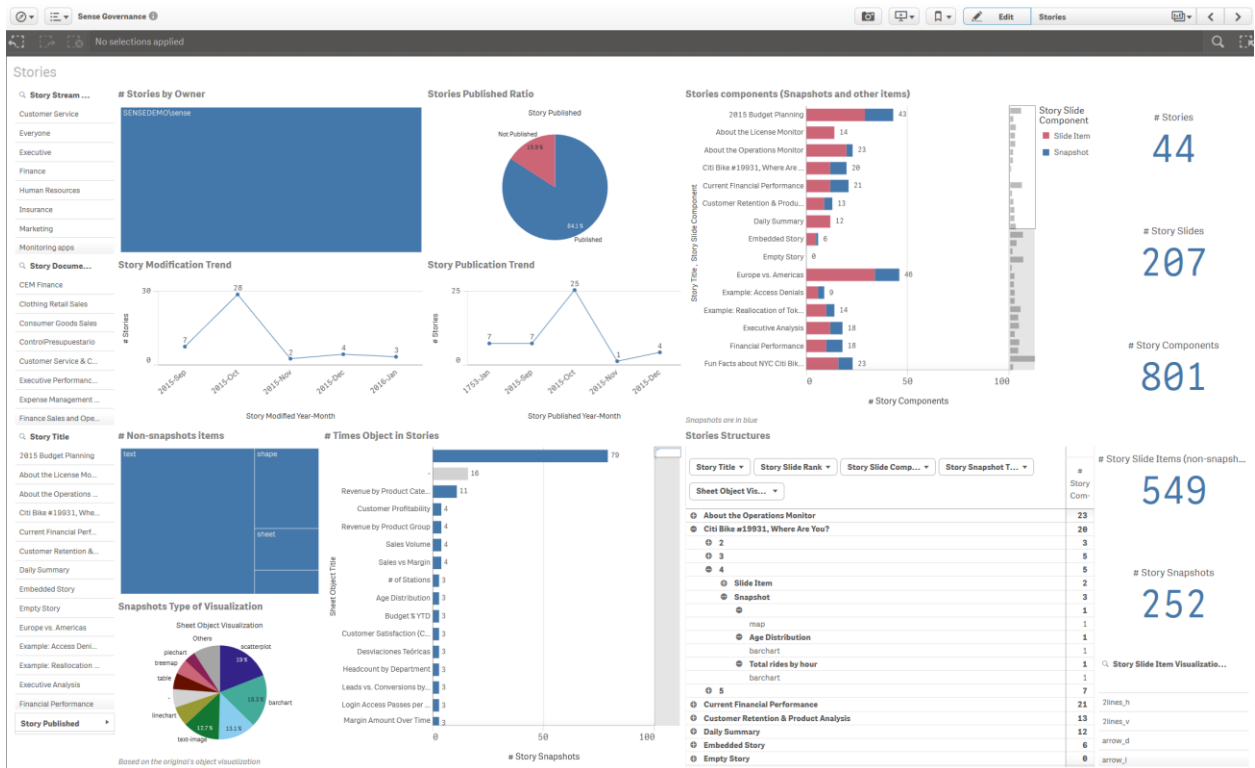
- Sheets – Who creates them? Are they published? What are the most used visualization types? Do all the objects have titles? If not, can I find the ones that should?



- Sheet Objects – How many objects do I have? What are their components in terms of dimension, measure, fields? Are they defined in the chart or using library items? Do I have everything labeled ok? Taking in account all the available fields of my datamodel, how many am I really using?



- Stories - Who are the owners of the stories? Are they published? What are the contents of the stories? How many snapshots are used and from which objects?



This draft is mainly focused in the creation of the indicators available in the library and the datamodel. The core of this proposal is “the field” because the initial idea was to be able to trace the impact a field has in its application(s). If I remove a field, what objects, measures, bookmarks, etc. would be affected?

As a next step, I believe this should be transformed/evolved in an application that could be easier to navigate and with a better choice of visualization to improve analysis. Also it would be very interesting to cross it with information that is available in the server logs, like session users vs. content owners.

Finally, the determination of which fields are used in measures is not foolproof yet, as it is merely comparing existing fields in the datamodel vs. “matching words” in the measure definition. The outcome is that a measure can have the correct field but also a few more matches. This is also something to look at for fine-tuning.

Try it yourself!

[Download here the Sense Governance.qvf](#) file and see how your applications are in your Qlik Sense Server.

This is pretty straightforward to use. Just make sure to have the following Data Connection:

- GovernanceQVDs – this is the same connection needed for the Sense Governance QVD Generator.qvf application.

Feedback

I hope you can try this out and please feel free to give me some feedback. Suggestions, solution proposals and critics are more than welcome!

You can reach me in <https://github.com/rvspt>