Putting the API in PowerBI

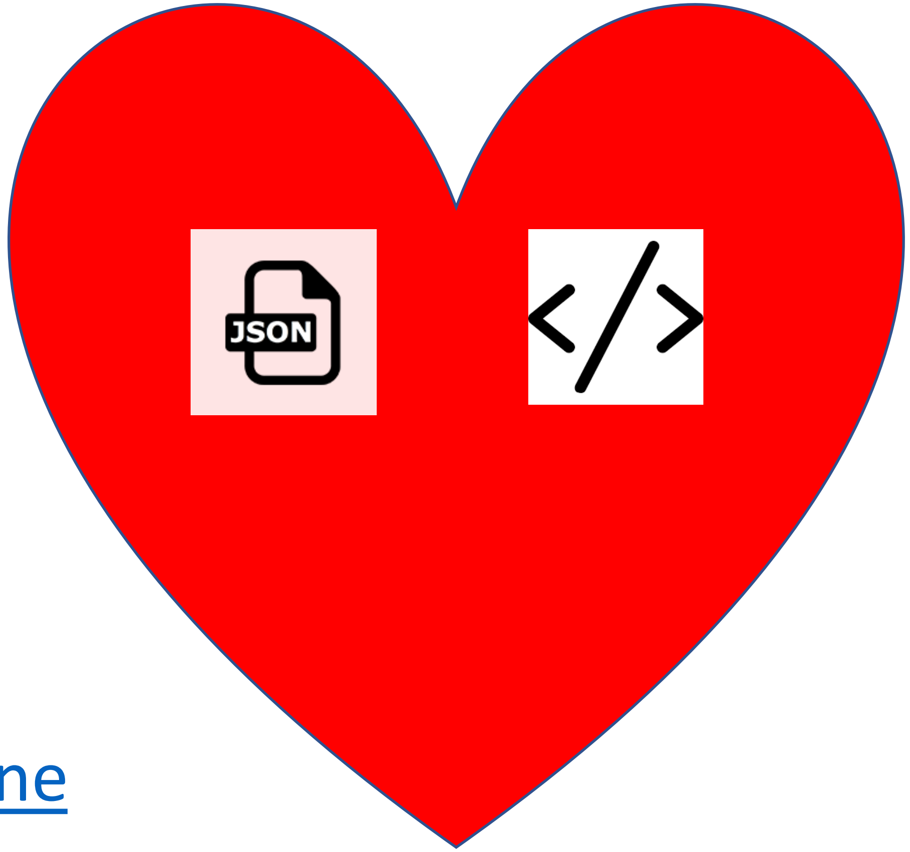
Jan Van Meirvenne

Topics

- Introduction
- Roleplay
- Putting the database to REST
- How to tell/discover what your API can do?
- Creating API connectors
- Connectors in the cloud
- Next steps
- Thanks&Takeaways

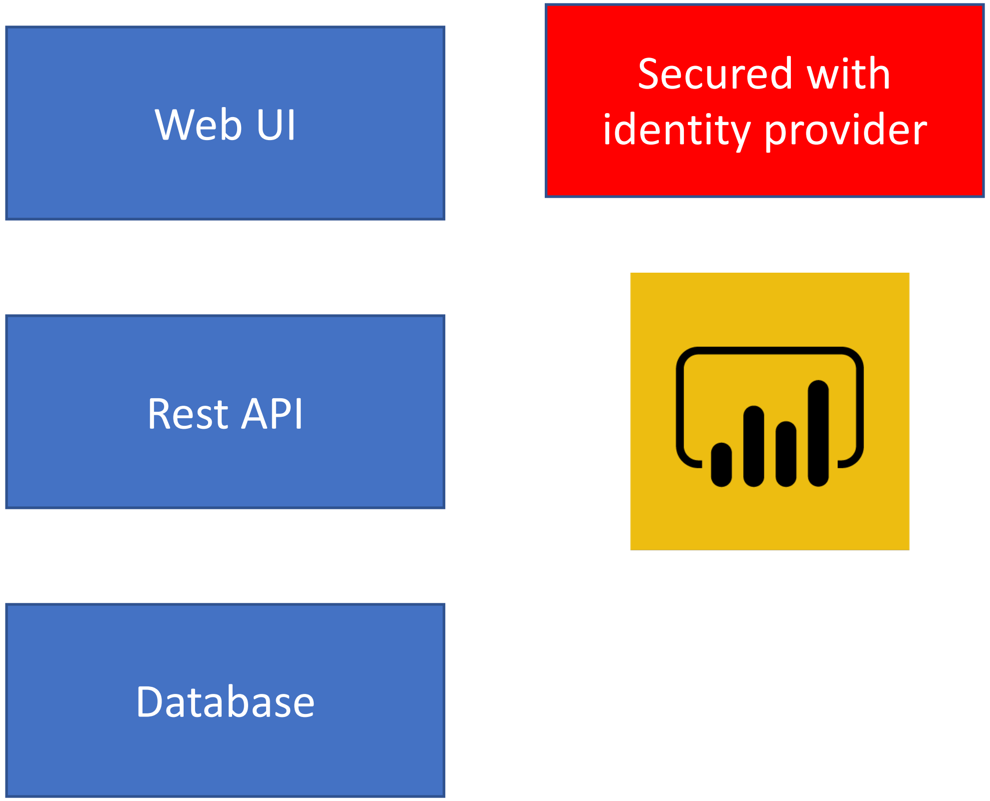
About Me

- Jan Van Meirvenne
- Technical Architect @ DexMach
  - Application Modernization
  - Cloud Data Engineering
  - Doing the things I love while getting paid
- Twitter: @AppDataDev
- Github: https://github.com/JanVanMeirvenne
- I don’t Instagram ☺



The next FaceBook: ToDo App

- It’s a todo application… you create to do’s …. for things you need to do
- Owned by ToDo .Inc
- Members:
  - Bob, the developer
  - Bill, the engineer
  - Bart, the analyst
- Goal: PowerBi as standard BI solution



Web UI

Secured with identity provider

Rest API

Database

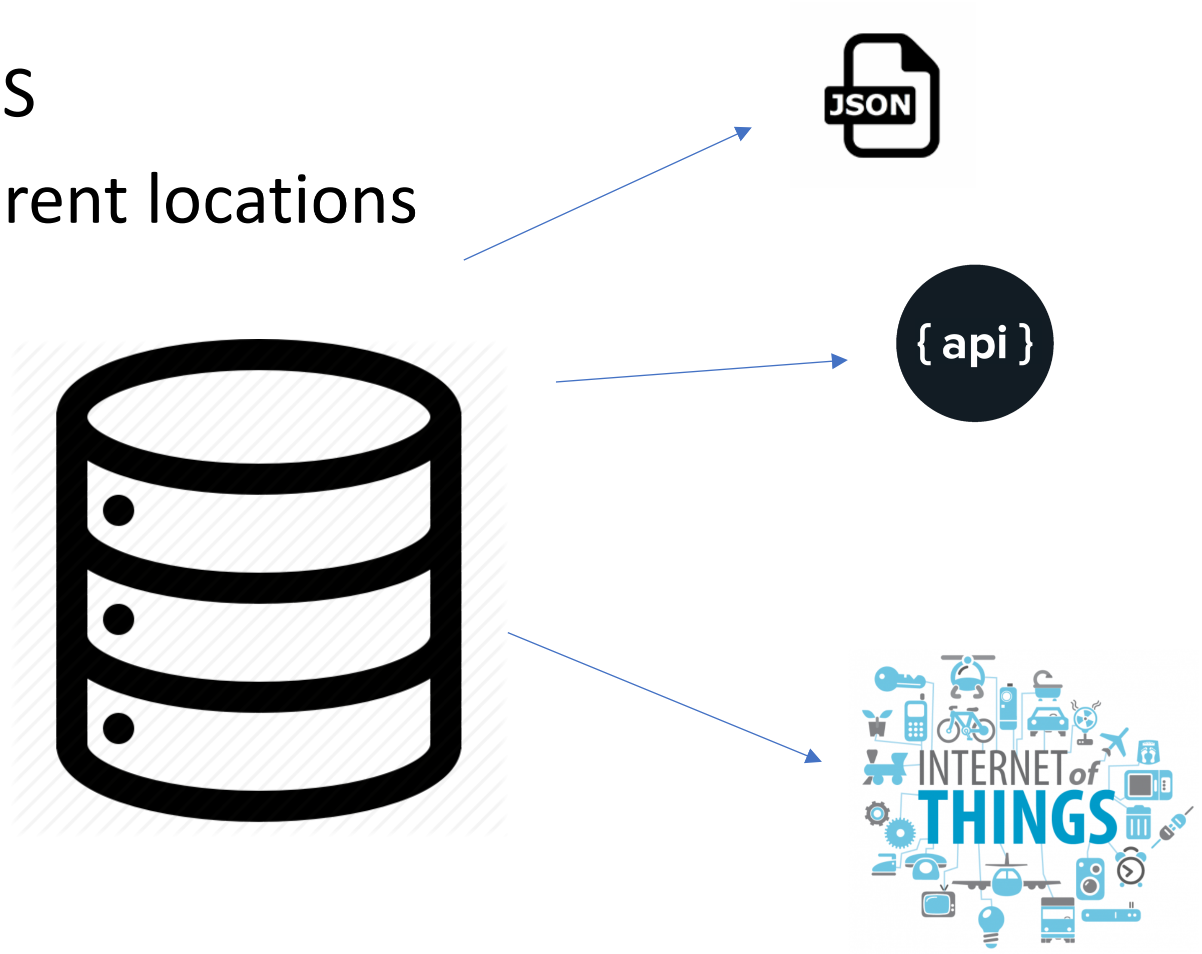
Congratulations, you get to preview free of charge

- Generate some test data AKA interactive session!

- Go to https://jvm-todo-web.azurewebsites.net
- Sign up (use mail option)
- Generate some (non-explicit) todo’s
- Lets see it in action further down the session!

Modern data

- Shift from IaaS to PaaS/SaaS
- Many data sources on different locations
- Secured endpoints
- Big & Unstructured Data



Where to start?

- Identify data source and way to extract it

- Components (which part of the application can expose data?)
  - The API interface allows us to extract data while adhering to the security
  - The database is not publicly accessible

- Contracts (which data can be queried how?)
  - Documentation
  - Tooling (eg SWAGGER: automatic generator of REST specs & clients)

Data for all

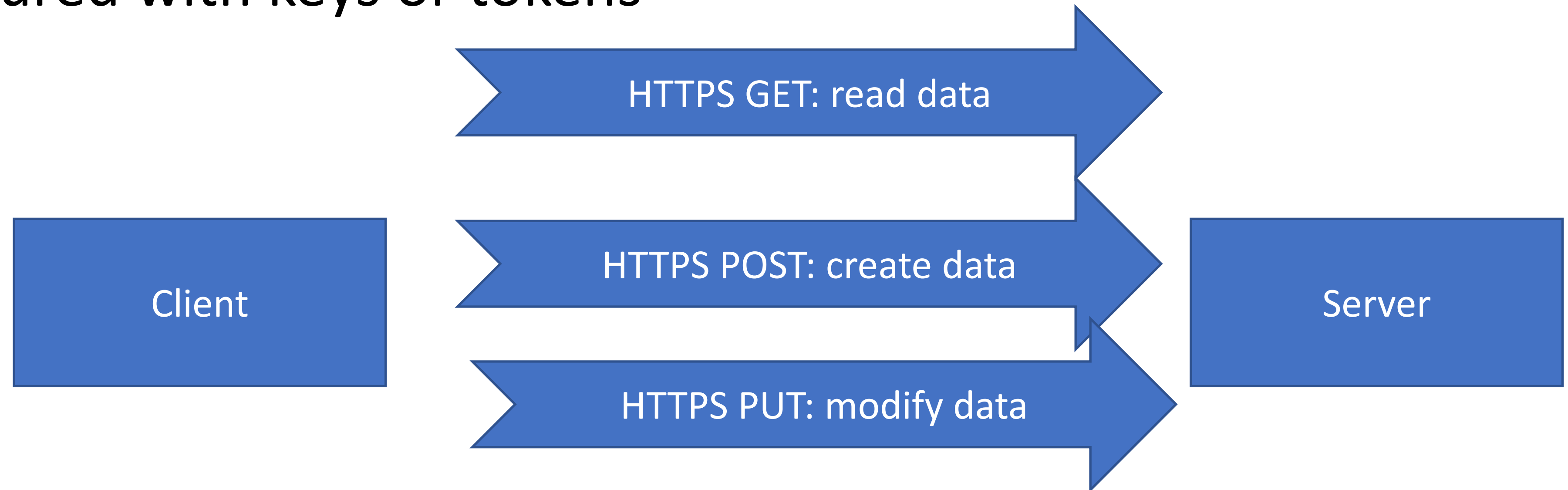
- Data engineering is very technical
- Analysts just need data, but don’t care where it comes from
- PowerBi connectors allow for drag n drop experiences

Generate Access Token
Query the API
Format JSON
Boilerplate PowerBI code
Invent nuclear fission

Hey Dude, here is data, go play!

What is REST?

- Representational State Transfer
- Runs on HTTP protocol
- Usually uses JSON as transfer format
- HTTP Methods to define intended action
- Secured with keys or tokens

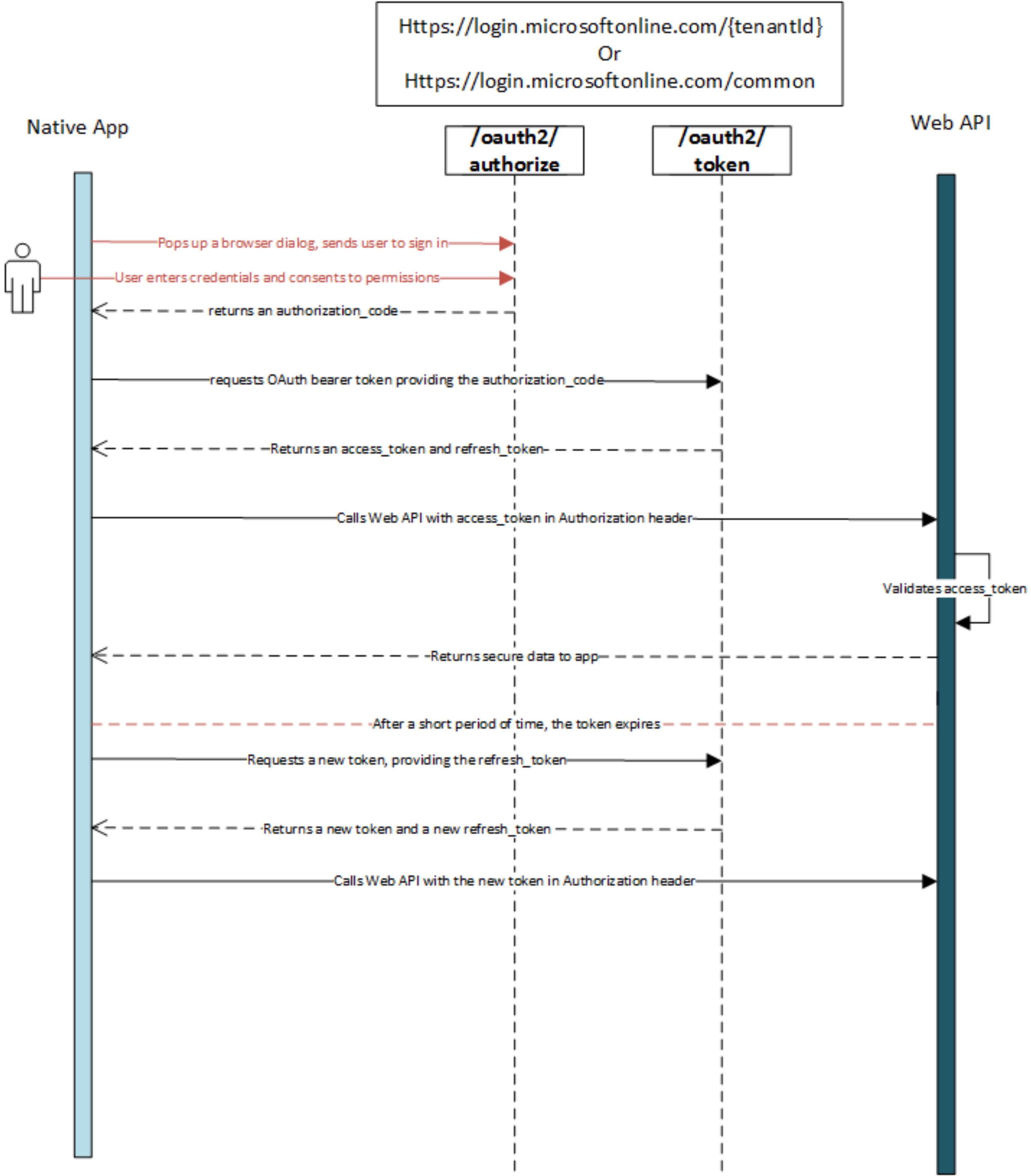


oAuth Flow

3 components
Client
Server
Authority

1) Client requests authority for “code” for access to Server
2) Authority grants code
3) Client exchanges code for an access_token and a refresh_token (silent session refresh)
4) Client uses access_token to access server. If token expires, refresh_token is used to renew

Can be made granular with roles/scopes



Https://login.microsoftonline.com/{tenantId}
Or
Https://login.microsoftonline.com/common

Native App

/oauth2/ authorize

/oauth2/ token

Web API

Pops up a browser dialog, sends user to sign in

User enters credentials and consents to permissions

returns an authorization_code

requests OAuth bearer token providing the authorization_code

Returns an access_token and refresh_token

Calls Web API with access_token in Authorization header

Validates access_token

Returns secure data to app

After a short period of time, the token expires

Requests a new token, providing the refresh_token

Returns a new token and a new refresh_token

Calls Web API with the new token in Authorization header

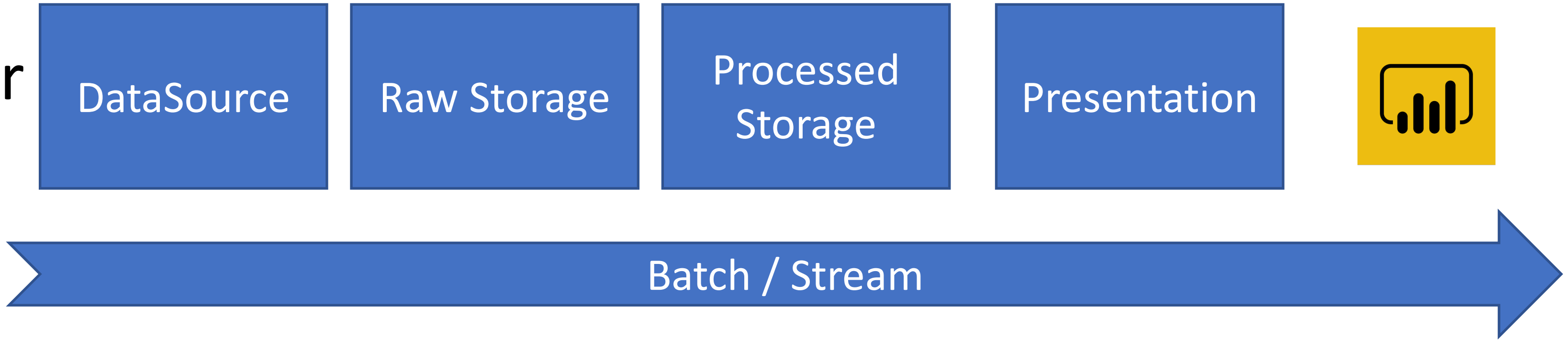
From prototype to baby steps

- PowerBi Desktop is a means to an end, but not the end
- Experimentation leads to analytical model, which needs central access
- We need PowerBi online!
- Disparities between Desktop & Cloud, will our connector survive?

From baby steps to world peace

- Data grows…non-stop
- Retention of data in the REST interfaces is limited

- Data Lake allows for the central storage of various data sources
- ETL flows can pull, organize and store data
- PowerBi will access the lake directly, or via a presentation database (DW)
- Data lake is the master



DataSource

Raw Storage

Processed Storage

Presentation

Batch / Stream

Key Takeaways

- Shift in data landscape
- Lots of API’s with interesting data
- Tools and Design can increase API discoverability
- PowerBI connectors can make API interaction effortless & secure
- Cloud can offload local development and increase collaboration
- Data lake as scale-up solution

Resources, should you be hungry for more

- Full codebase on Github (will add PowerPoint): https://github.com/JanVanMeirvenne/ToDoApp
- https://jvm-todo-web.azurewebsites.net

- Swagger: https://swagger.io (recommend Nswag for .NET)
- PowerBi Connector SDK: https://github.com/microsoft/DataConnectors
- Lambda Architecture (for modern data ETL): http://lambda-architecture.net (also check out Kappa for full-streaming approach)
- OAUTH2/JWT for API security: https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba

Thanks

- Now I will take my REST ☺
- and someone else can do the REST
- All these bad jokes why?
- Only so that I can close the session with saying…API

- Note to self: nightly session building promotes bad humor