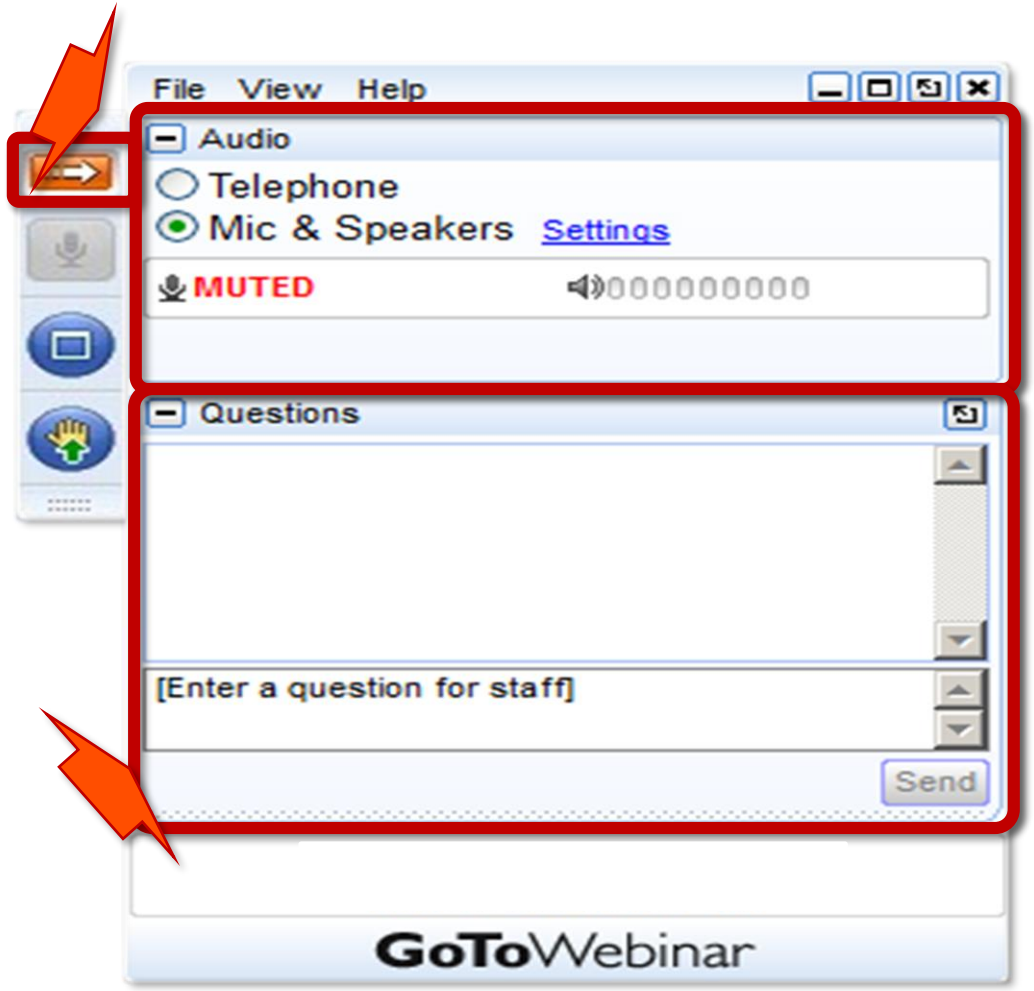
OData: What’s New with REST APIs for Your Database

Sanjeev Mohan, Gartner

Nishanth Kadiyala, Progress

Mark Biamonte, OData TC Member, Progress

Audio Bridge Options & Question Submission



OData: What’s New with REST APIs for Your Database

Sanjeev Mohan, Gartner

Nishanth Kadiyala, Progress

Mark Biamonte, OData TC Member, Progress

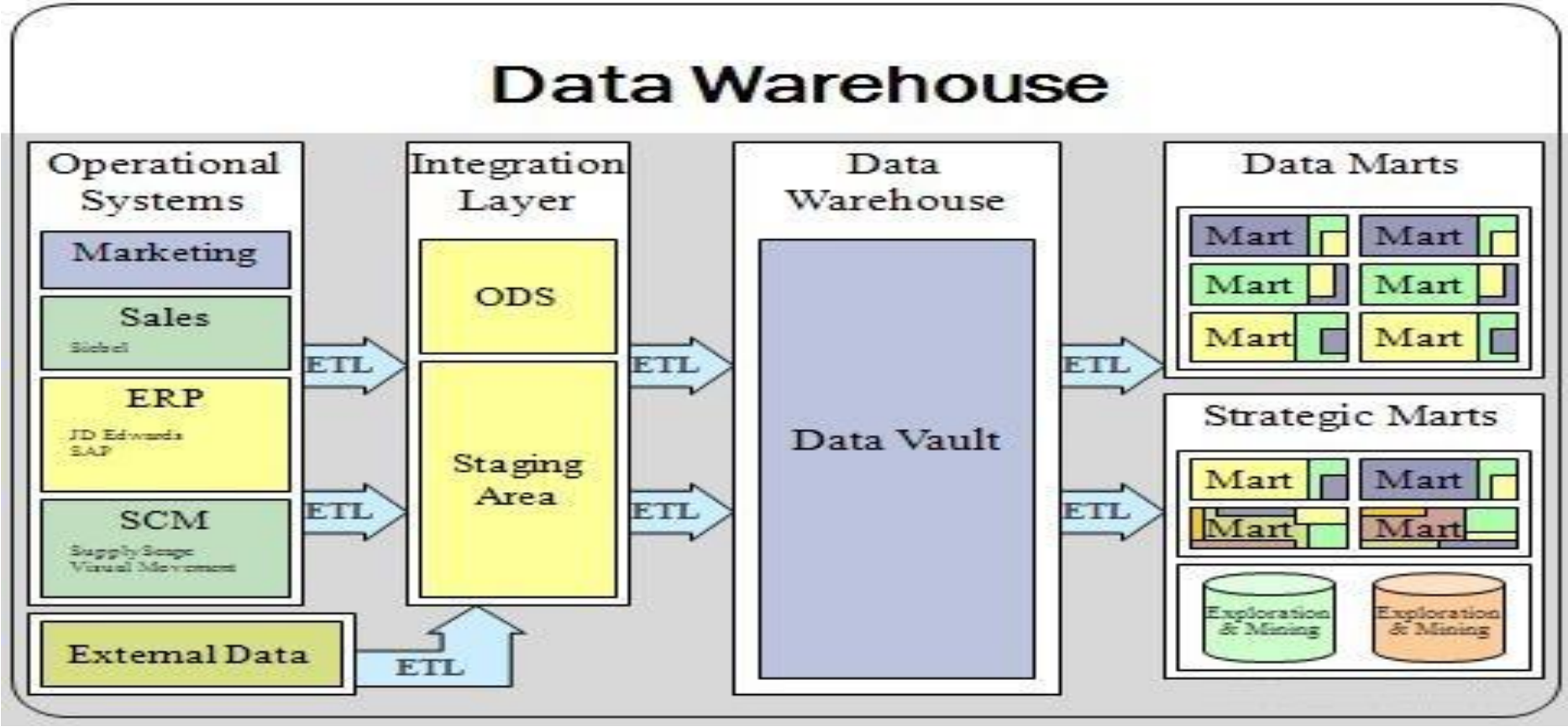
Agenda

- Modern Data Access Patterns
- Why are companies RESTifying databases?
- OData overview
  - Exposing the Database Using OData
  - Getting Started with OData
  - Demo

Progress®

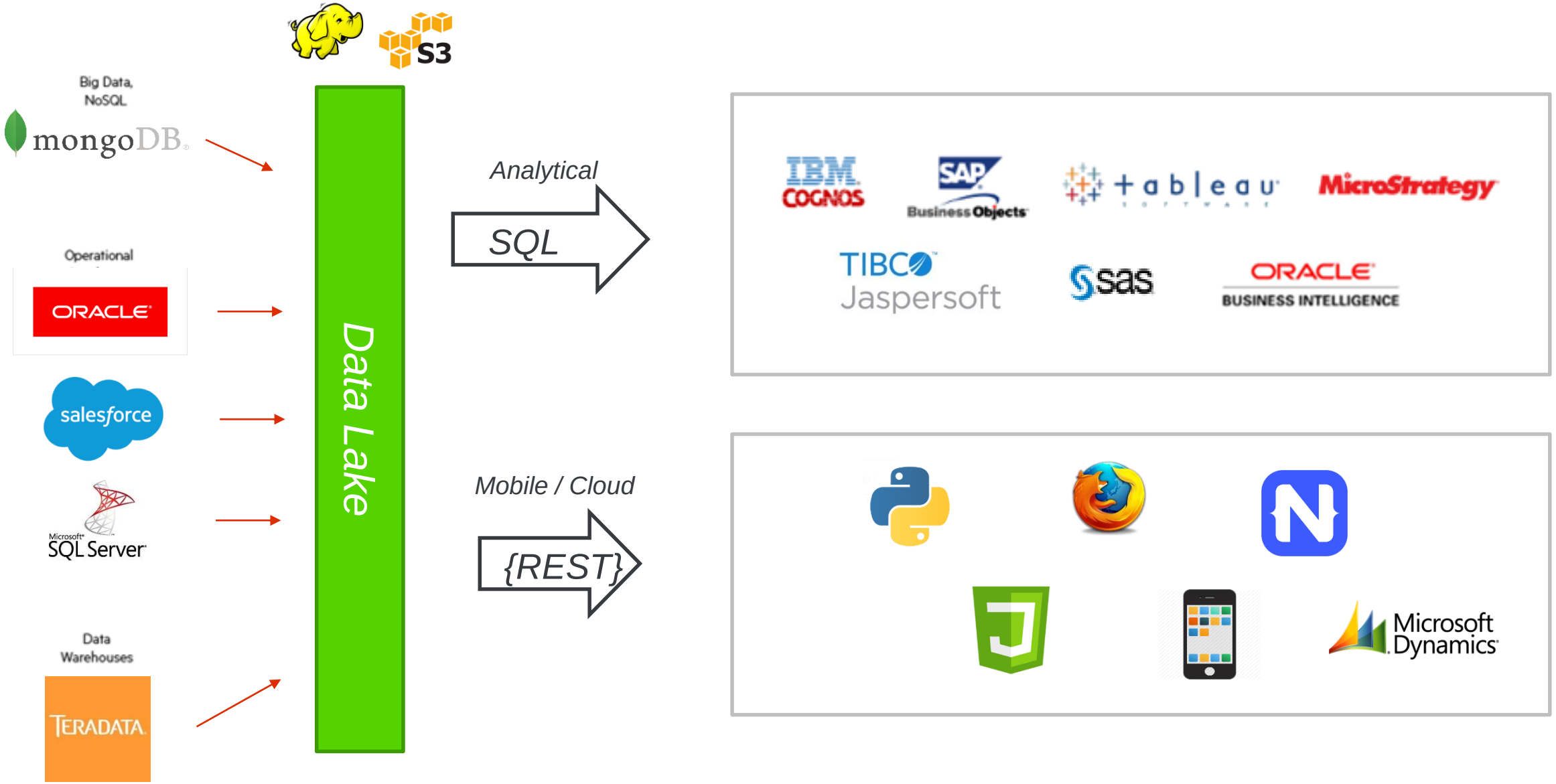
Modern Data Access Patterns

Data Warehouses – still the workhorse analytical engines

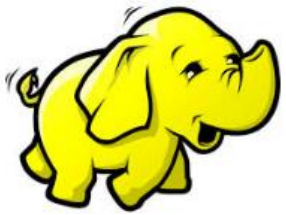


Source: https://en.wikipedia.org/wiki/Data_warehouse

Data Access for Data Lakes

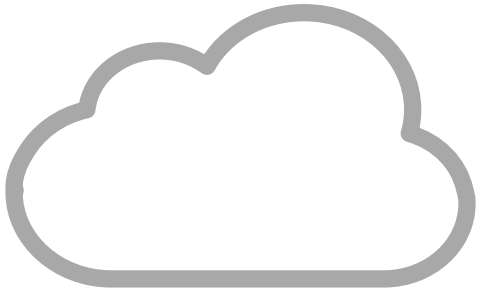


Common Data Lake Implementation Technologies Create Differing Requirements for Data Governance



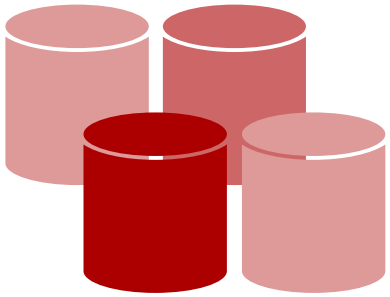
Hadoop distributions:

- *Simplified data ingestion and storage with several processing options*
- *Data lake management ecosystem emerging*
- *Complex deployment and management*



Cloud-based block and object stores:

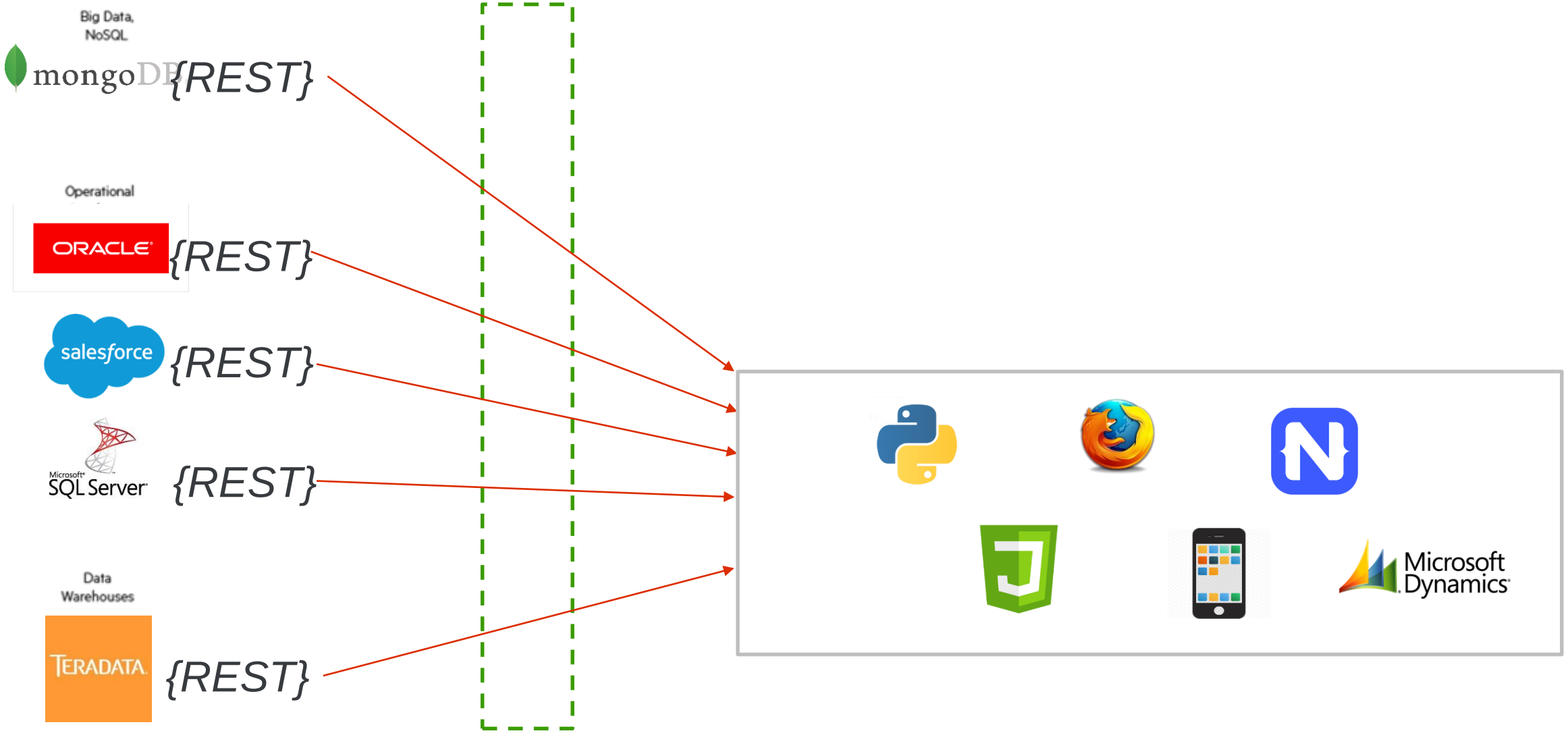
- *Simplified data ingestion and storage*
- *Bring your own processing*
- *Nascent management and security ecosystem*



Database management systems:

- *Optimal for certain data types and formats*
- *Data processing options expanding beyond SQL*
- *Scaling and cost may be challenges*

Virtual Data Lake by REST Enabling Sources



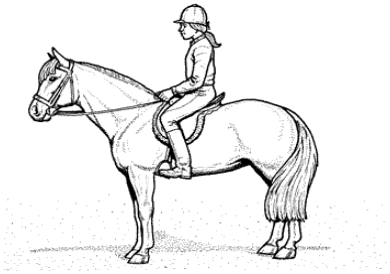
Goal is to “Connect” data and not “Collect”

- REST APIs are accessible over the internet and thus are much more user/coder friendly
- REST is stateless and thus can easily be horizontally scaled since there is no session dependency
- As you monetize data, APIs offer a secure way to expose your data to users outside your organization
- REST APIs are more human-readable and thus more engaging
- Decoupling the code from the infrastructure will give the enterprise the flexibility to adopt newer technologies

Why are companies RESTifying databases?

Transition from SOAP to REST

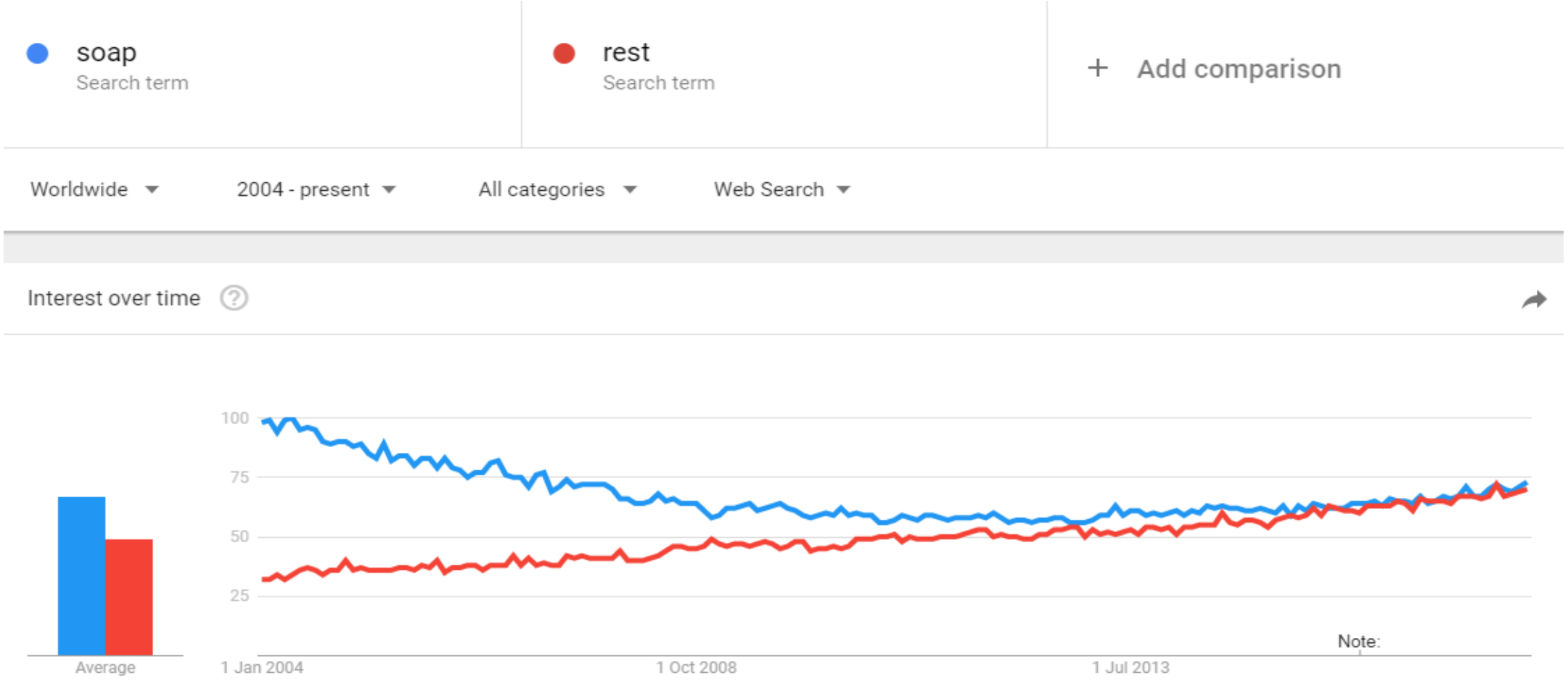
SOAP vs REST



SOAP
XML
Uses WSDL to define interface
Stateful

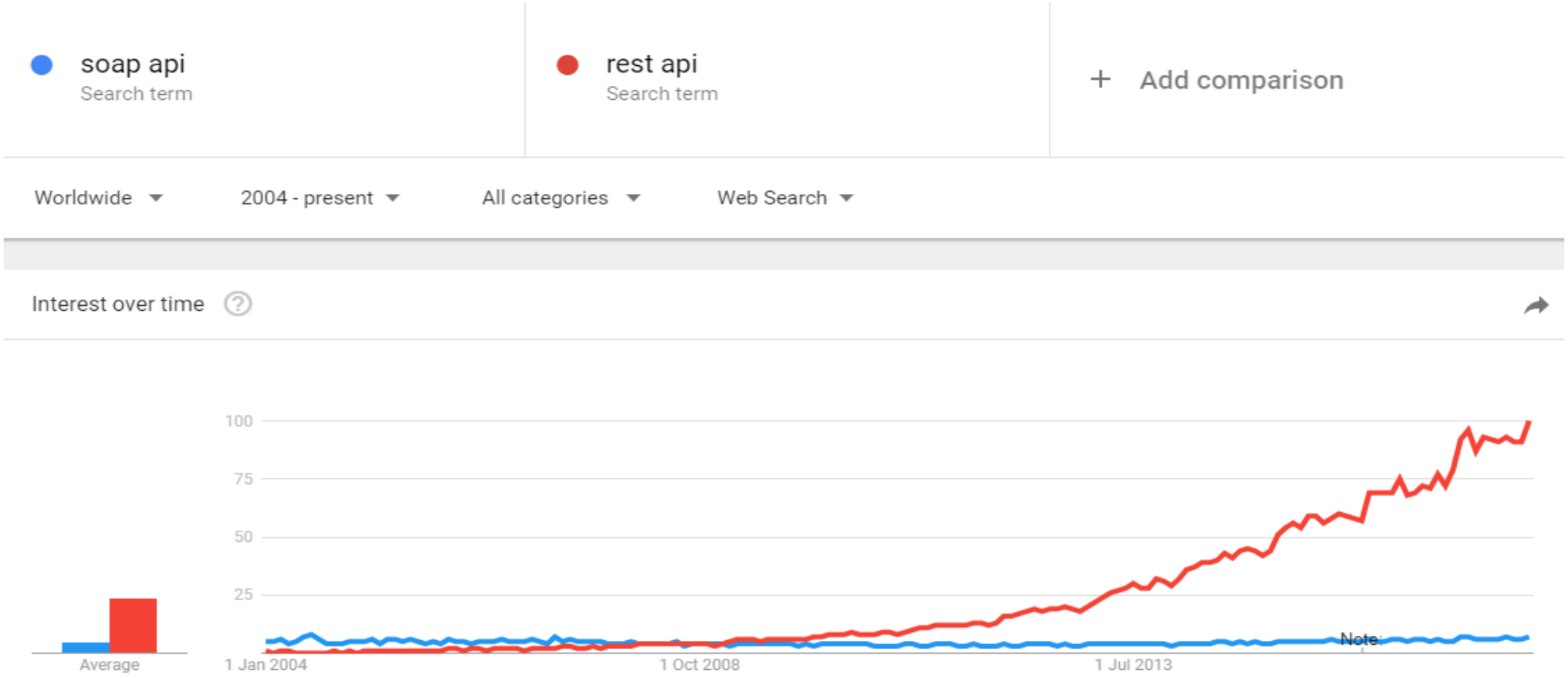
REST
JSON, XML, YAML etc.
Uses HTTP verbs (Get, Post, Put, Delete)
Stateless

SOAP vs REST



Source: Google Trends

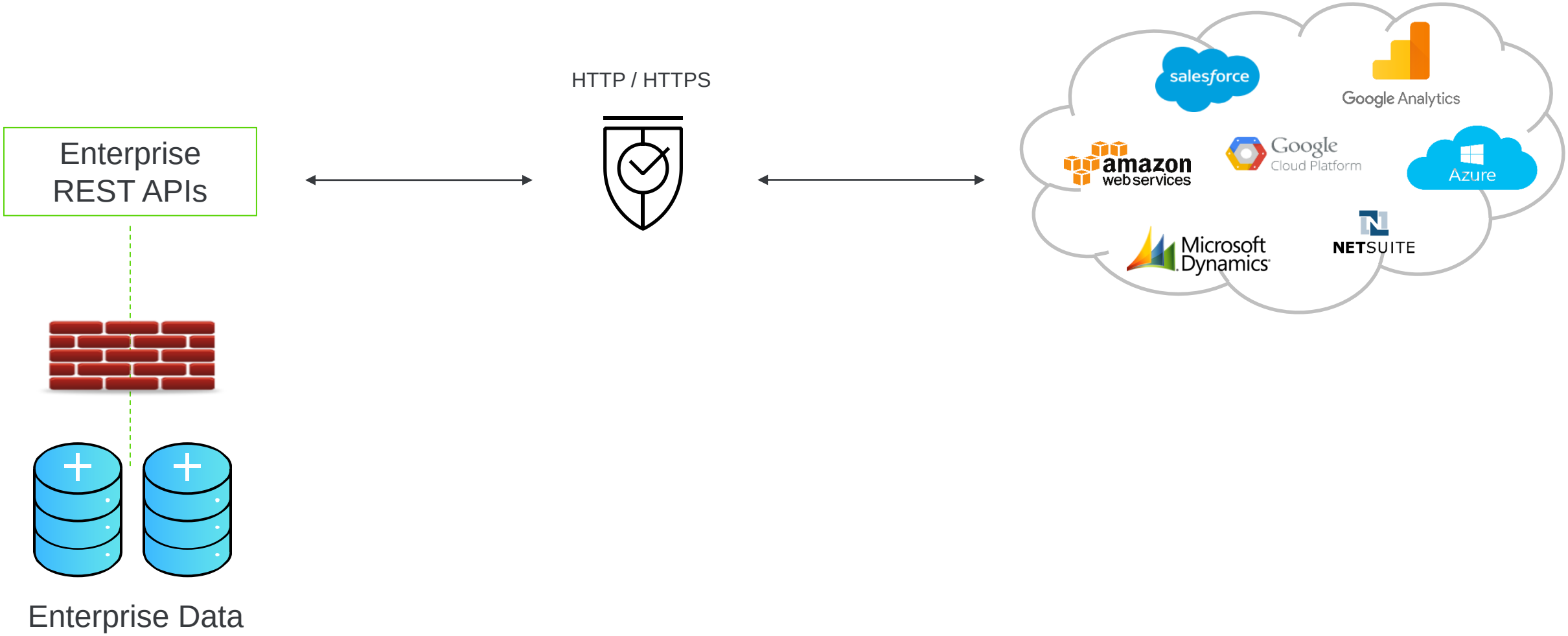
SOAP vs REST



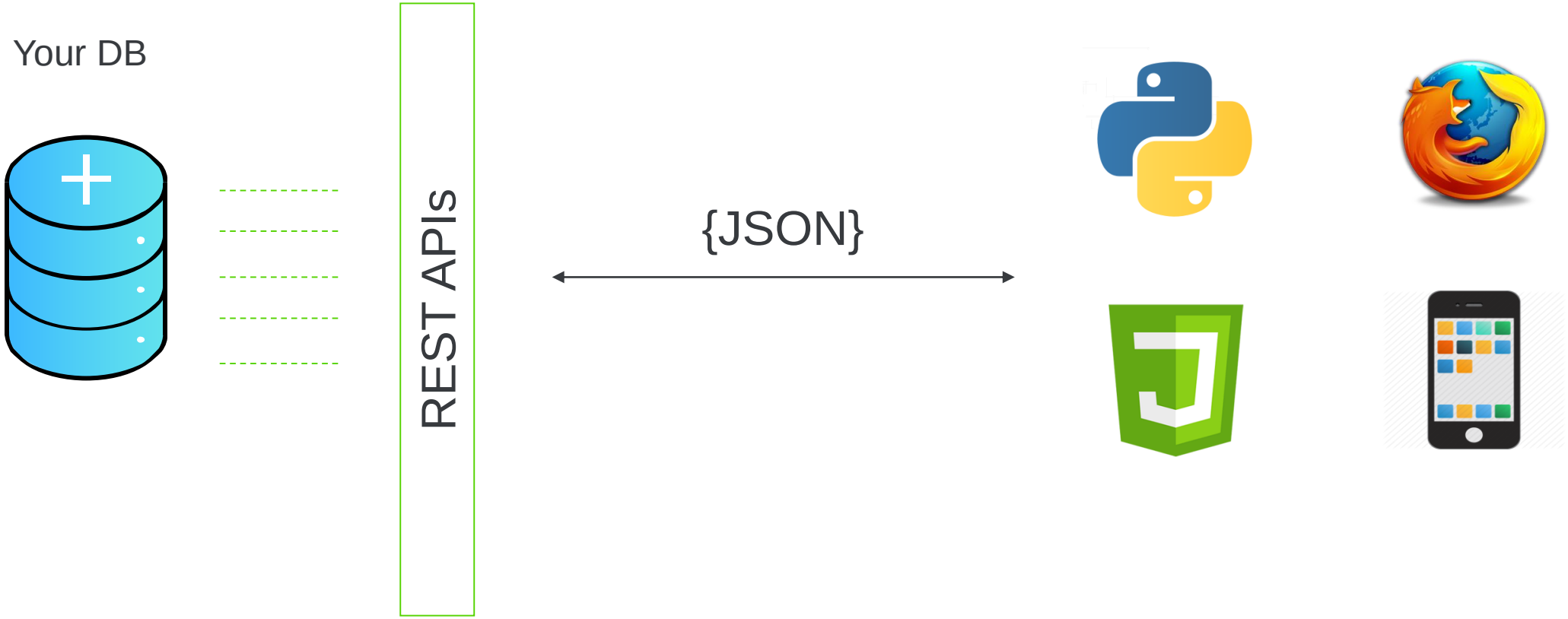
Source: Google Trends

Use Cases for REST Enabling Databases

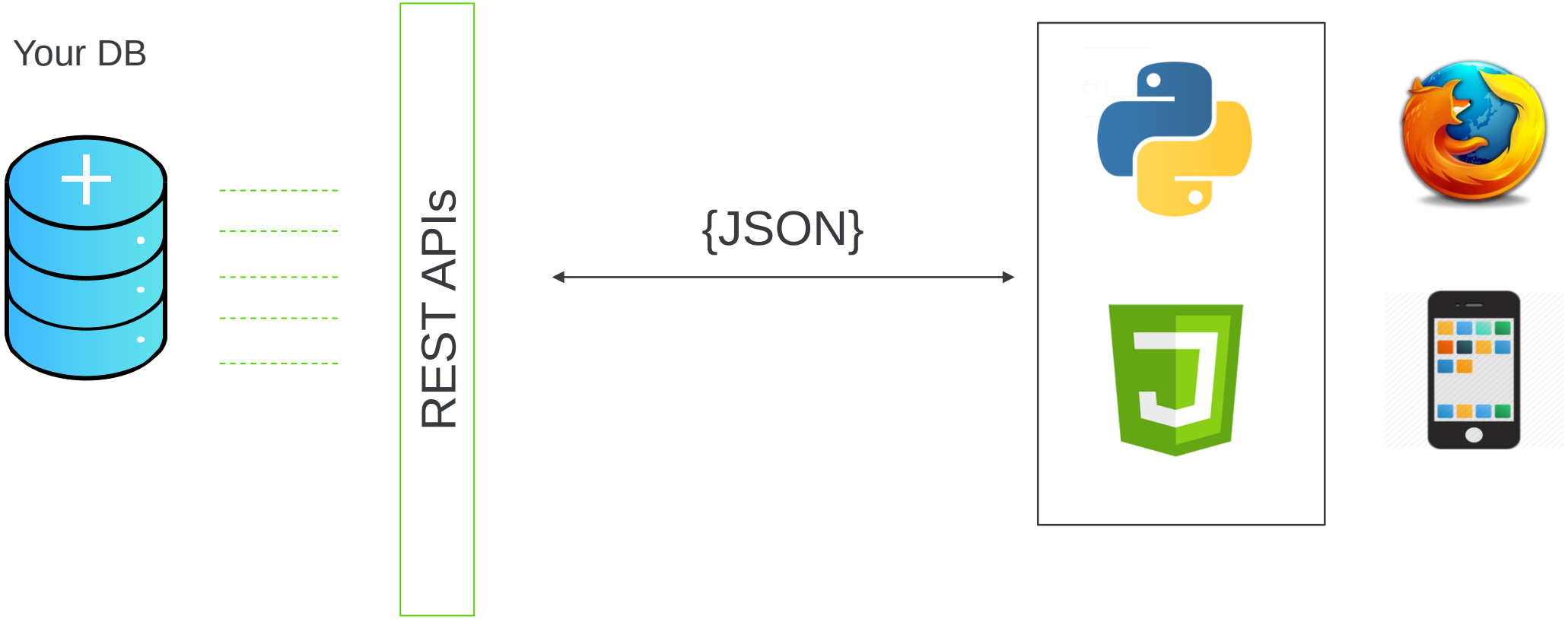
1. Expose Enterprise Data to SaaS/Cloud via HTTP



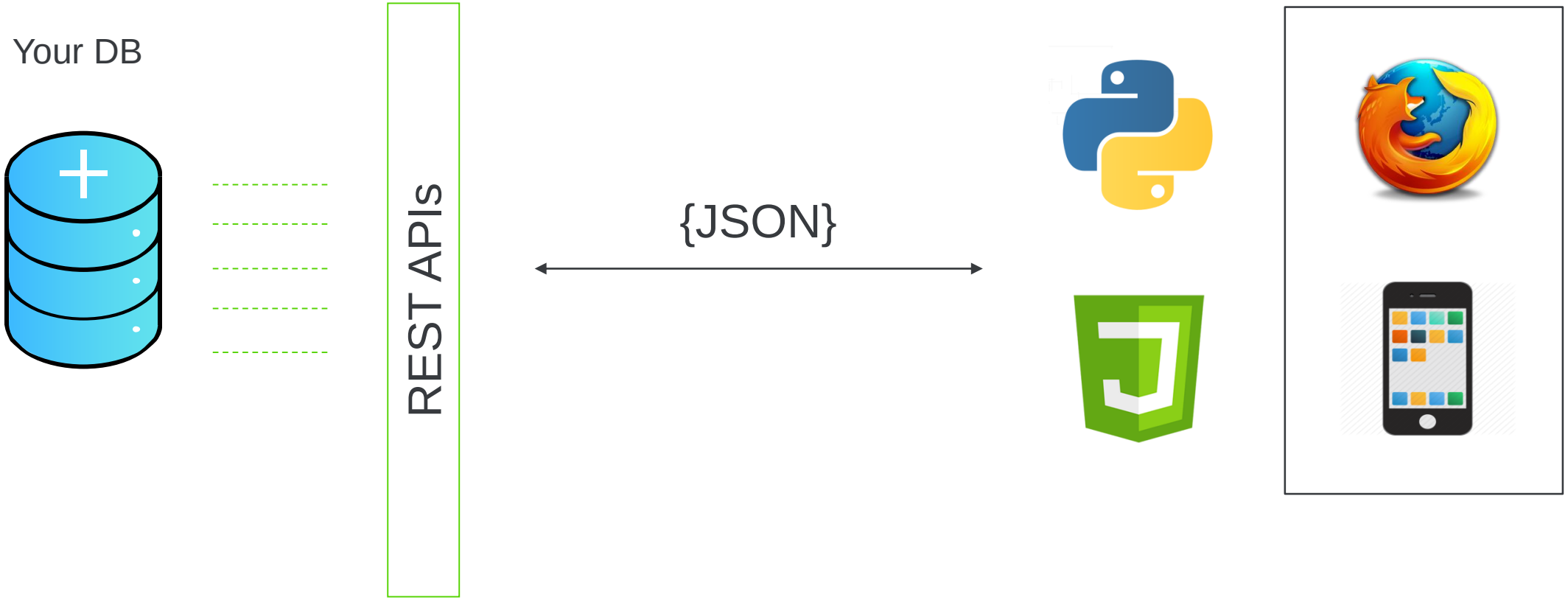
2. Modernize Access to Enterprise Data



2. Modernize Access to Enterprise Data



2. Modern Data Access



3. Additional Layer means Additional Control

Limit direct access to your DB

Throttling can help protect against resource monopolization

Decoupling code from the database makes future transitions easier

Poll Question 1

Which of the following use cases are true for your organization?

- Your database needs to support modern languages such as JS, node.js, angular, python, etc.
- Sensitive data needs to stay within the firewall (GDPR, etc.)
- Enterprise Data needs to be available from new devices
- Modern Apps need REST end points for integration
- Other

Industry is moving towards REST APIs

Cloud-Hosted Databases

Google Cloud

Azure

Database.com

IBM Cloudant

…

On-Premises Databases

Oracle REST Data Services

Teradata REST API

IBM DB2 Rest Services

MarkLogic REST API

…

Poll Question 2

Which of the following databases are of interest to REST enable?

- Oracle
- SQL Server
- IBM DB2
- Postgres
- MySQL

Poll Question 2 – Part 2

Which of the following databases are of interest to REST enable?

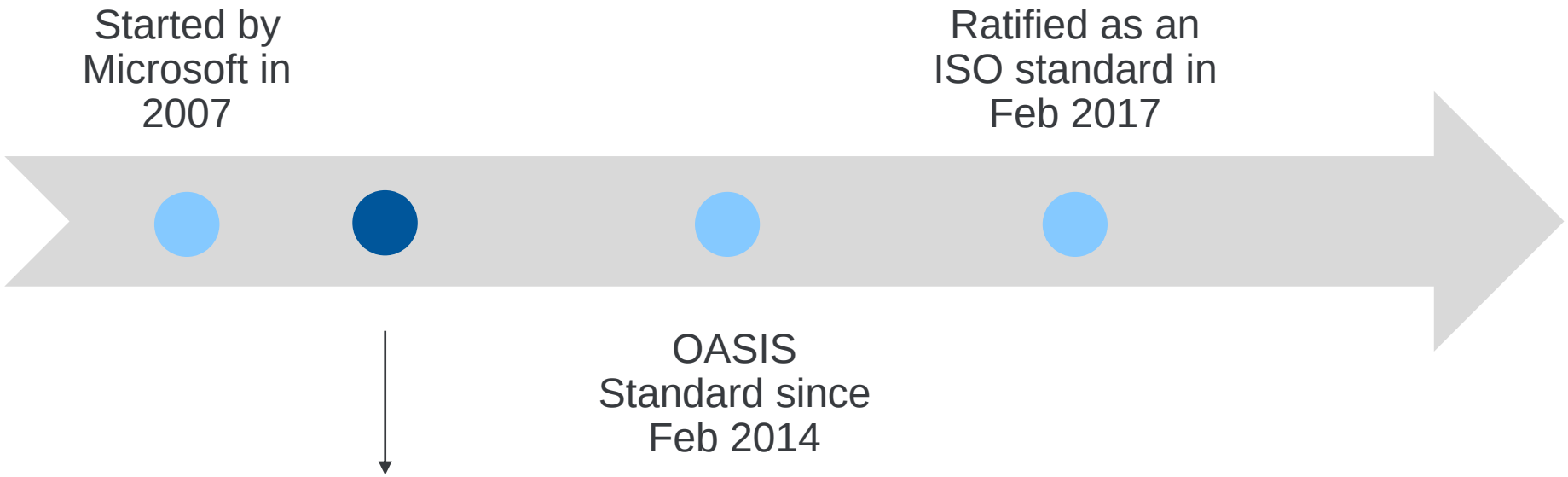
- Amazon Redshift
- Hadoop Frameworks
- Teradata
- OpenEdge
- Other

But, every database and REST API is different… OData is the Answer

OData Overview

What is OData?

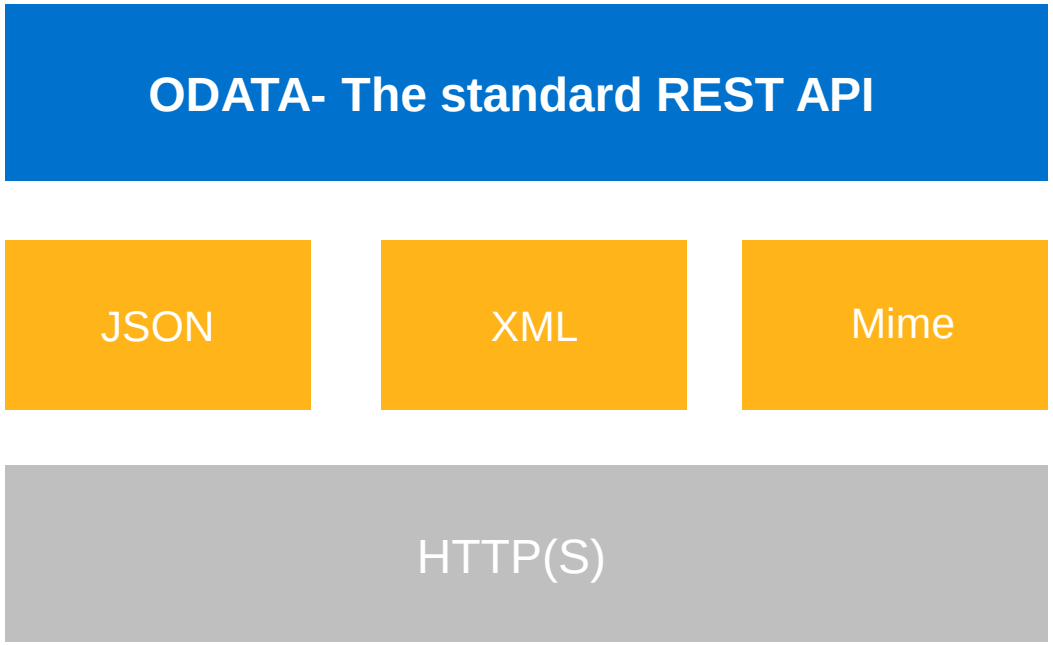
An open protocol to allow the creation and consumption of queryable and interoperable RESTful APIs in a simple and standard way



Progress was the first member to join the OData Technical Committee

OData is the standard for REST

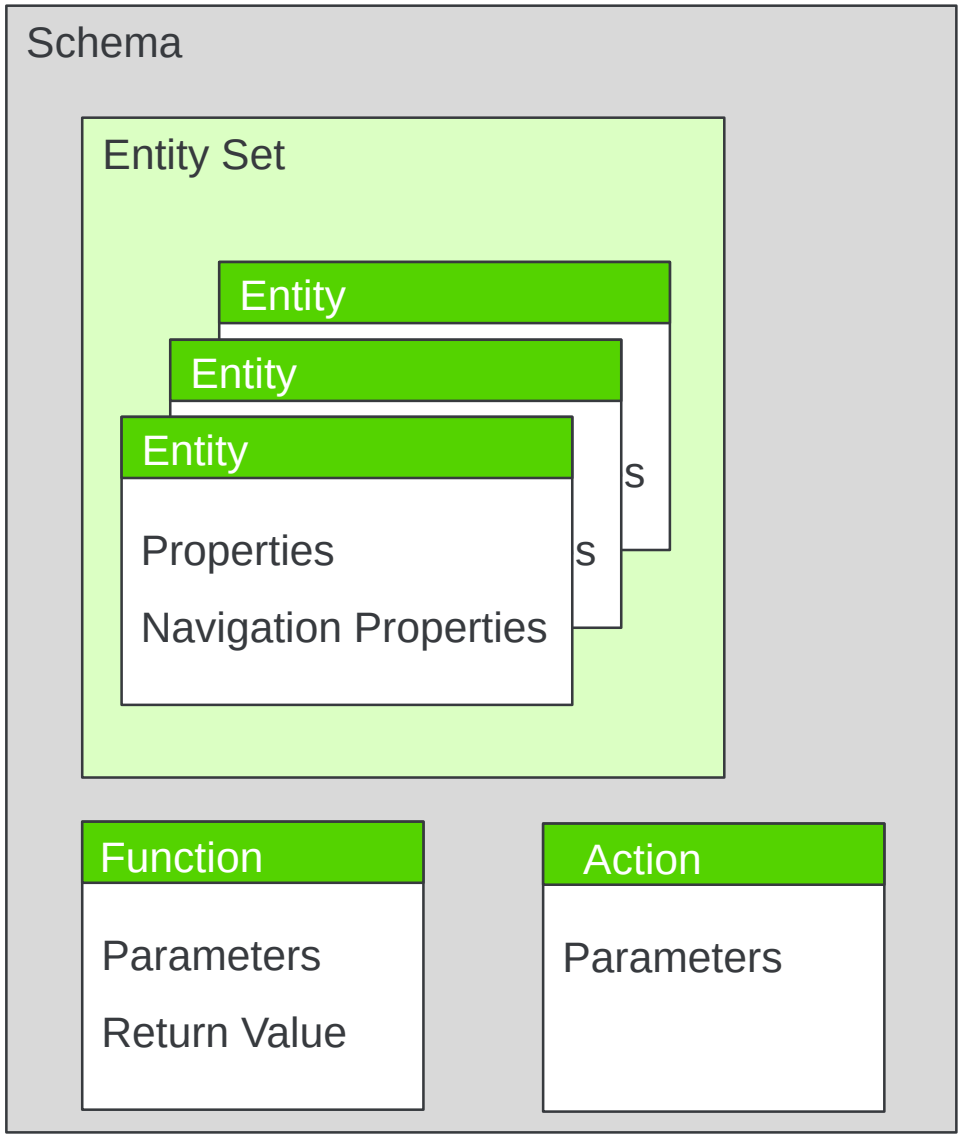
OData is essentially SQL for the web built on top of standard protocols – HTTP, JSON & ATOM – while leveraging the REST architecture style

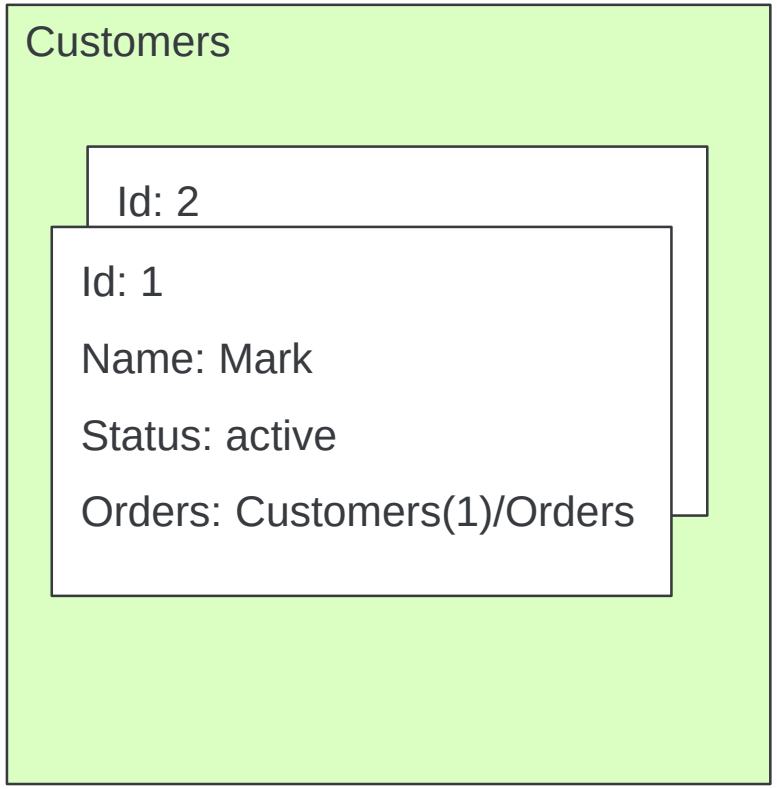


Broad Adoption for OData

Exposing the Database Using OData

OData Data Model





Mapping Databases to OData

Database Construct	OData Construct
Table	Entity Set
Row	Entity
Column	Property
Stored Procedure	Action (or Function)
User Defined Function	Function
Primary Key	Entity Key
Foreign Key	Navigation Property

Mapping SQL to OData

HTTP Method	OData REST Operation	SQL Command
POST	Create a new entity	INSERT
GET	Retrieve zero or more entities	SELECT
PUT	Update an Entity (replace semantics)	UPDATE
PATCH	Update an Entity (update semantics)	UPDATE
DELETE	Destroy an Entity	DELETE
GET	Invoke a function and get its results	CALL / EXEC
POST	Invoke and action and get its results	CALL

Self Describing Service – Metadata Document

- Detailed description of the service

URL: <service_root>/$metadata

Entity Type Complex Type	Function Action
Type Properties	Parameters
Property Attributes	Parameter Attributes
Name,Type, MaxLength, Precision, Scale, Nullable, more	Name,Type, MaxLength, Precision, Scale, Nullable, more
Navigation Property	Return Type (Function)
Key (EntityType)	Type, Nullable
	Function Import Action Import

- Information equivalent to SQL Tables, Columns, Primary Key, Foreign Key, Procedures, ProcedureColumns, Functions, FunctionColumns metadata calls

URL Query Conventions

```
https://myodataserver/northwind
\_____________________________/
                |
          Service Root
```

```
https://myodataserver/northwind/Customers('ALFKI')/Orders?$expand=OrderDetails
\_____________________________/\_________________________/\____________________/
                |                          |                        |
          Service Root               Resource Path            Query Options
```

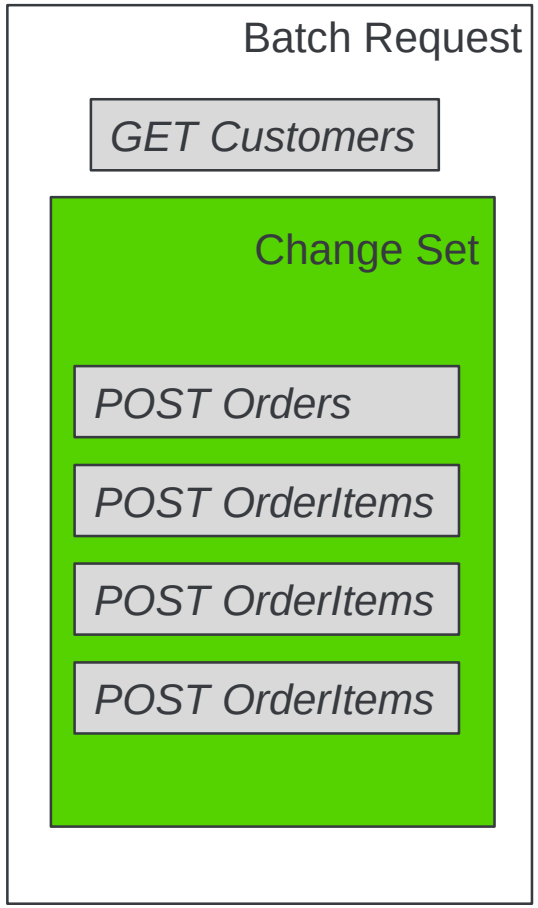
ODATA Query Options

System Query Option	Description	SQL Construct
$filter	Restrict the entities returned when querying an entity set to those matching the filter criteria	WHERE clause
$select	Specify the properties to be included in the returned entities	SELECT list
$orderby	Specify the sort order of the returned entities	ORDER BY clause
$expand	Include related entities and complex types nested in the returned entities	INNER JOIN
$top and $skip	Enable client to page through results	TOP/SKIP or LIMIT/OFFSET
$count	Include the count of the number of entities returned in the result	COUNT(*)
$search	Restrict the entities returned when querying an entity set to those matching the search expression	Full Text Search
$format	Specify the desired data format for the response	N/A

Batch Insert and Transactions

URL: <service_root>/$batch

- Execute multiple operations in a single OData request.
- Operations in a Change Set are executed atomically

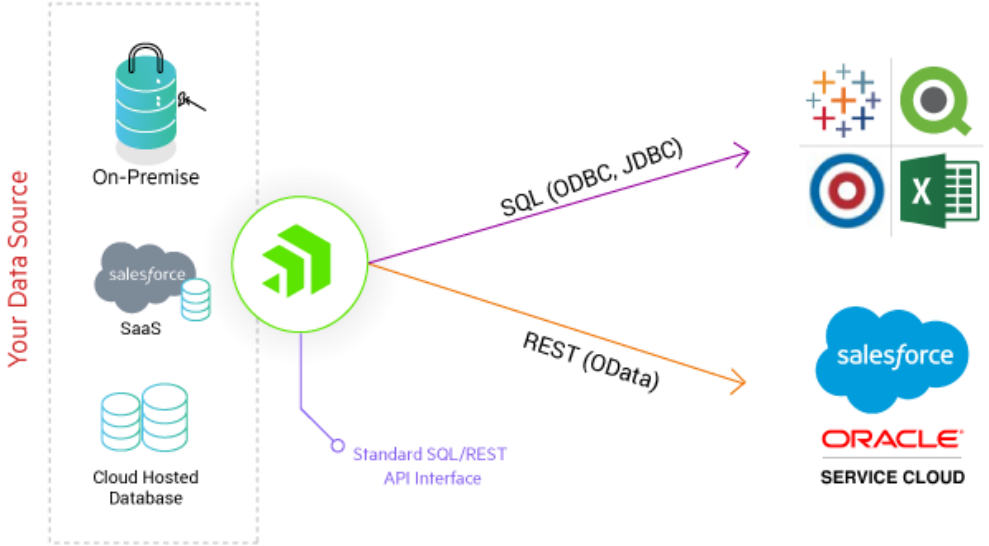


Getting Started with OData

OData Libraries - http://www.odata.org/libraries/

.NET	Java	JavaScript	Other
RESTier	Apache Olingo	Node-odata	ODataCpp (C++)
ODataLib	SDL OData Frameworks	DevExtreme	Pyslet Python Package
Edmlib	Odata4j	o.js	ODataStore for CoreData (iOS)
ASP.NET Web API OData	Jello Framework	OpenUI5	OData4ObjC (iOS)
AdaptiveLINQ	ODataJClient	JayData	OData Client Library for Tcl/Tk
Microsoft.Spatial		Breeze.js	

Or use Hybrid Data Pipeline to create OData REST API from your enterprise databases



Why Hybrid Data Pipeline?

- OData from any data source
- No Coding
- No version control needed
- Patented on-premises gateway technology

Demo

Resources

- Understanding OData in 6 steps: http://www.odata.org/getting-started/understand-odata-in-6-steps/
- Getting Started Tutorial on OData.org: http://www.odata.org/getting-started/basic-tutorial/
- Getting Started Tutorial with OData for your enterprise databases: https://www.progress.com/tutorials/odata/rest-api-for-sql-server-oracle-or-postgres-via-odata
- REST API Debate across OData, GraphQL, ORDS: https://dzone.com/articles/rest-api-industry-debate-odata-vs-graphql-vs-ords
- FAQs we hear on OData: https://www.progress.com/blogs/odata-faqs-why-should-rest-api-developers-use-odata
- [Webinar] OData external data strategies for SaaS vendors with guests from Microsft and Oracle: http://forms.progress.com/wbr-odata-integration-for-saas
- OData examples across marketing APIs: https://www.progress.com/blogs/getting-started-with-rest-apis-across-marketo-eloqua-and-odata
- Hybrid Data Pipeline Product Page: https://www.progress.com/cloud-and-hybrid-data-integration

Progress®

Appendix

Query Examples

Service Root: http://myodataserver/northwind

Resource	
Customers	http://myodataserver/northwind/Customers
$select=x,y	http://myodataserver/northwind/Customers?$select=CompanyName,CustomerID,Country
$filter	http://myodataserver/northwind/Customers?$filter=Country eq 'Germany'
$orderby	http://myodataserver/northwind/Customers?$orderby=Country,City desc
$expand	http://myodataserver/northwind/Customers?$expand=Orders
$top and $skip	http://myodataserver/northwind/Customers?$orderby=Country,City desc&$top=5&$skip=5
$count	http://myodataserver/northwind/Customers?$count=true

Putting it altogether

```
http:// myodataserver/northwind /Customers?
    $expand=Orders($select=OrderID,EmployeeID,OrderDate;$filter=EmployeeID eq 4)
   &$filter=Country eq 'Germany' and (City eq 'Berlin' or contains(City,'Frankfurt’))
   &$select=CustomerID,CompanyName,ContactName,City,Country,Phone
   &$orderby=CompanyName desc
   &$count=true
```