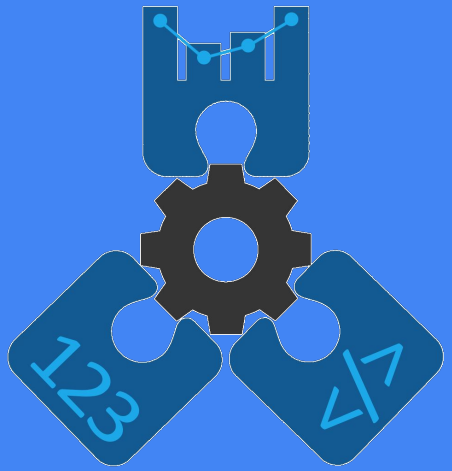
metadata technology
NORTH AMERICA



Rich Data Services

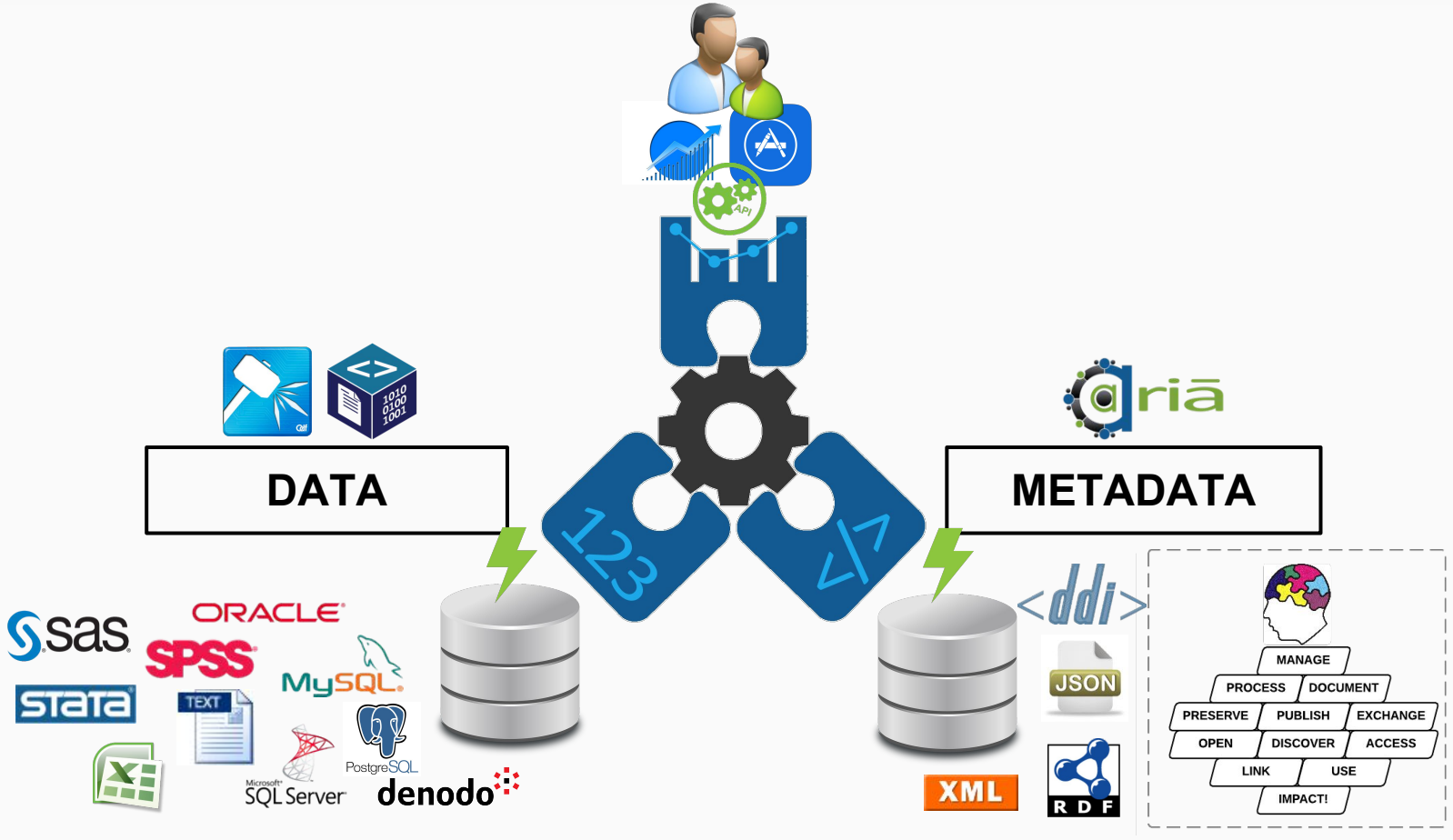
Making Data Human and Machine Friendly

http://www.mtna.us | mtna@mtna.us

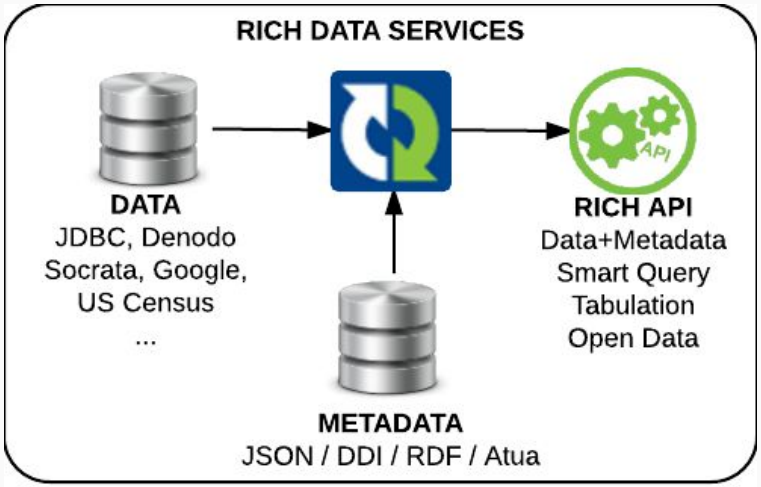
How do we want to access data?

- Static access/download for offline analysis (full or partial datasets) in easy to reuse formats (open data)
- Dynamically query to browse/discover/explore
- Tabulation / aggregate
- Consumable as a services: API/JSON, etc. (statistics as a service)
- Delivery metadata with the data
  - Data Product = Data + Metadata
- Above often seen as different needs solved by different systems
- Can we make this a unified solution?

Rich Data Services



Rich Data Services (RDS)



- Brings data / metadata together
- Realize the vision of statistics as a service
- RDS is a middleware REST API
- Back end queryable data sources (SQL)
- REST based services / operations

- Deployed between your data sources and users/applications
- Knowledge about the data captured and stored in a local repository
- Powerful API delivers your data as a service, complemented by comprehensive metadata, innovative querying features, and flexible serialization options

RDS features

- /select operation
  - Enhanced SQL like query capabilities
  - Flexible field selection (incl/excl, regex, keywords/concepts)
  - Derivation / on the fly recodes
  - Metadata injection (alongside data)
  - Row/Column paging (large/wide tables)
- /tabulate operation
  - Dedicated service for aggregation (dimensions, measures)
- Flexible serializers for easy consumption / integration
  - RDS JSON: full features
  - Popular JSON: Google Charts, amCharts, plotly, Denodo,...

RDS Features

- Metadata services (/catalog, /variables, /classifications)
  - Can **search** and access detailed information on catalog, collections, data sources, variables, classifications, etc.
  - Supports **incremental metadata enhancements** (start with schema, edit in UI, bulk upload JSON or DDI)
  - **Repository** options: JSON, OrientDB, Atua/Ariā, (Colectica?)
  - Server side **metadata inference**, profiling, analysis agents
- Open Data Packaging service (/package)
  - Large query/tabulation can be "packaged" for delivery
  - SlegdeHammer on the web (wrapper)
  - Self-service data shop (async order processing)

Knowledge Discovery Agents

- Complement manual editing or bulk loading
- Reduce burden of capturing metadata
- Discover knowledge in the data
- Run automatically server side and contribute to metadata
- On load or on demand
- Inference / Profiling: data type / representation, range, uniqueness / primary keys, ...
- Analysis: descriptive statistics, missing values, classification discovery, statistical distribution, disclosure risk, machine learning?

RDS Features

- Queryable Data Sources
  - SQL/JDBC: MS-SQL, MySql, MonetDB, Oracle, PostgreSQL, Vertica, Denodo (DV), ...
  - Socrata (http://www.socrata.com) (15K+ public datasets)
  - Google BigQuery (https://cloud.google.com/bigquery/)
  - More planned (Google Fusion, US Census API, etc.)
- Miscellaneous
  - Admin UI (catalog, metadata, config/ monitoring)
  - Service throttling
  - Integrate w/WSO2 middleware for advanced API control, SSO
  - Disclosure Control (planned)

RDS Use Cases

- *For public, private, or internal use*
- Data / Statistics as a Services
- Open data access / packaging
- Catalogs / Data portals
- Data Analysis & Visualization
- Provide access to Big Data (subsets, back-end engines)
- Self-service facilities
- Data request management
- <insert your use case here>

Availability / Early Adopters Program

- RDS planned for release 3Q/4Q 2016
- Flexible licensing options: on premises, dedicate/shared cloud
- You can get access to the technology today
  - Present an exciting use case!
  - Willingness to evaluate and work with us
  - Some level of commitment to product
- Benefits:
  - Product will fit your needs first
  - Licensing discount
  - Lead innovation

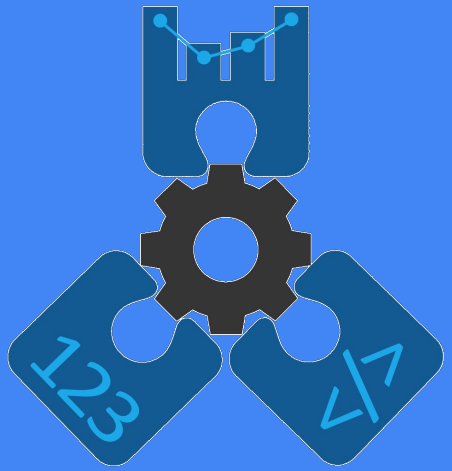
RDS Summary

- Realize the statistics as a service vision
- Queryable data products (data+metadata)
- RDS is a metadata driven query / tabulation / packaging engine
- Enable dynamic / personalized data delivery
- Both for internal/external use
- Open Data Developer friendly
- Offer self-service environment
- Secure
- Reduce burden, consumable data, reuse
- Combines and unifies other management components

Demo Data Sources

- American National Election Study 1948: MySql, 67 vars / 662 obs
  - http://www.electionstudies.org/studypages/1948prepost/1948prepost.htm
- US Census 2000 5% PUMS: MonetDB, 113 vars / 6,257,697 obs & 163 vars / 14,271,294 obs
  - https://www.census.gov/census2000/PUMS5.html
- MIDUS 3. MonetDB, 2,575 vars / 3,294 obs
  - http://midus.wisc.edu/midus3/index.php
- Montgomery County: Socrata, 12 vars / 9,100 obs
  - https://data.montgomerycountymd.gov/Human-Resources/Employee-Salaries-2014/54rh-89p8
- USA Names: Google BigQuery, 5 vars / 5,552,452 obs
  - https://cloud.google.com/bigquery/public-data/usa-names

metadata technology
NORTH AMERICA



Rich Data Services

Use Cases / Examples

http://www.mtna.us | mtna@mtna.us

RDS: select variables by keywords or access detailed metadata

/select?cols=**$truman**&rowLimit=10&metadata

Truman Variables

Our data deals with the 1948 American Elections when Truman was running against Dewey. Lets do a keyword search for "Truman" with the select API and `cols` parameter to select all the variables that have to do with Truman.

http://prod.mtna.us:8080/rds/api/test/NES1948/select?cols=$truman&rowLimit=1(Run

Data

V480014a	V480014b	V480015a	V480015b	V480031a	V480031b	V480031c
30	91	98	91	10	0	0
30	50	30	91	13	11	0
10	30	30	91	10	0	0
30	91	10	91	11	0	0
30	60	10	91	11	12	0
30	91	99	91	12	11	0
98	91	90	91	0	0	0
50	90	90	91	10	12	0
50	30	30	90	10	0	0
30	90	90	91	0	0	0

Variable Metadata

Variable metadata can be accessed through the `variables` and `variable` API.

All Variables

We can select all variables using the `variables` endpoint. We can paginate the metadata using the `colLimit` and `colOffset` parameters, similar to the `select` API.

http://prod.mtna.us:8080/rds/api/test/NES1948/variables?colLimit=5 Run

V480013 - PRESLELCTN OTCM SURPRISE

Name	V480013
Label	PRESLELCTN OTCM SURPRISE
Format	NUMERIC 42
Classification	V480013

V480014a - WHY PPL VTD FOR TRUMAN 1

V480014b - WHY PPL VTD FOR TRUMAN 2

RDS: inject metadata to add meaning to variables and codes

Injecting Categorical Values

The metadata avaiable form the Rich Data Services makes it easy to display the data or categorical values for the data. Use the </> button to switch between the two.

http://prod.mtna.us:8080/rds/api/test/NES1948/select?rowLimit=5&metadata Run </>

Data

VVERSION	VDSETNO	V480001	V480002	V480003	V480004	V480005	V480006	V
1	1948.T	7218	1001	2	4	1	1	
1	1948.T	7218	1002	1	4	2	1	
1	1948.T	7218	1003	1	4	1	1	
1	1948.T	7218	1004	2	4	1	1	
1	1948.T	7218	1005	1	4	2	1	

Injecting Categorical Values

The metadata avaiable form the Rich Data Services makes it easy to display the data or categorical values for the data. Use the </> button to switch between the two.

http://prod.mtna.us:8080/rds/api/test/NES1948/select?rowLimit=5&metadata Run </>

Data

VVERSION	VDSETNO	V480001	V480002	V480003	V480004
1	1948.T	7218	1001	2. TOWN OR CITY	4. NAME NOT KNOWN
1	1948.T	7218	1002	1. METROPOLITAN AREA	4. NAME NOT KNOWN
1	1948.T	7218	1003	1. METROPOLITAN AREA	4. NAME NOT KNOWN
1	1948.T	7218	1004	2. TOWN OR CITY	4. NAME NOT KNOWN

RDS: choose different serialization options for immediate integration in popular frameworks or tools

Google Charts

Pie Chart

We can create an Google pie chart based on an aggregation of a categorical variable.

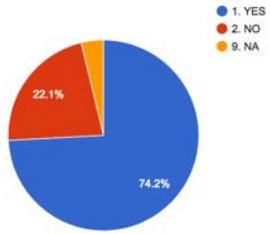
V480007

Data retrieved using:
`http://prod.mtna.us:8080/rds/api/test/NES1948/tabulate?dims=V480007&measure=count:COUNT(V480007)&format=gcharts&metadata`

Constructing the Pie Chart

By setting the `format` parameter to "gcharts" we retrieve the JSON as cols and rows that can be immediately plugged into the `chart.data` object.

Chart



Pie Chart

We can create an AMCharts pie chart based on an aggregation of a categorical variable.

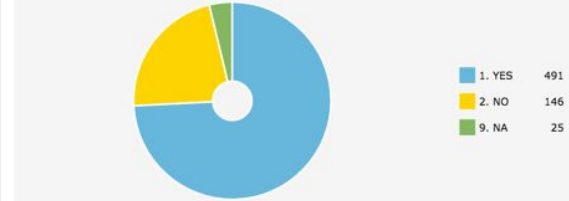
V480007

Data retrieved using:
`http://prod.mtna.us:8080/rds/api/test/NES1948/tabulate?dims=V480007&measure=count:COUNT(V480007)&format=amcharts&metadata`

Constructing the Pie Chart

By setting the `format` parameter to "amcharts" we retrieve the JSON as a dataProvider. This JSON object can be immediately plugged into the AMCharts JSON.

Chart



Pie Chart

We can create a Plotly pie chart based on an aggregation of a categorical variable.

V480007

Data retrieved using:
`http://prod.mtna.us:8080/rds/api/test/NES1948/tabulate?dims=V480007&measure=count:COUNT(V480007)&format=plotly_pie&metadata`

Constructing the Pie Chart

By setting the `format` parameter to "plotly_pie" we retrieve the JSON as a data object. This JSON object can be immediately plugged into the Plotly JSON.