Overview of Row Level Security (RLS)

How to restrict data access at the row level

Alex Cortez - Lead Solution Engineer - OEM/Embedded Analytics
Justin Craycraft - Principal Solution Engineer - OEM/Embedded Analytics

Agenda

- What is RLS?
- Two main types of RLS
- Entitlements deep dive
  - Demo - Deepest Granularity Entitlements
  - Demo - Sparse Entitlements
- How to use the RLS demo kit
- Q&A

What is Row Level Security?

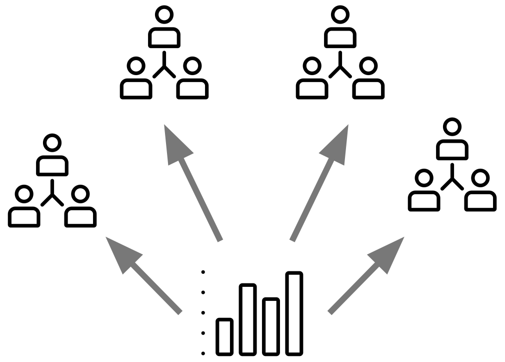
- Row Level Security (RLS) in Tableau refers to restricting the rows of data a certain user can see in a published workbook/data source
  - For ex: People managers working with HR data should only be able to see data records related to employees on the team they manage

Sensitive Data

Compliance Requirements

REQUIREMENTS
RULES
LAW
STANDARDS
COMPLIANCE
GOVERNANCE
POLICIES
REGULATIONS
TRANSPARENCY

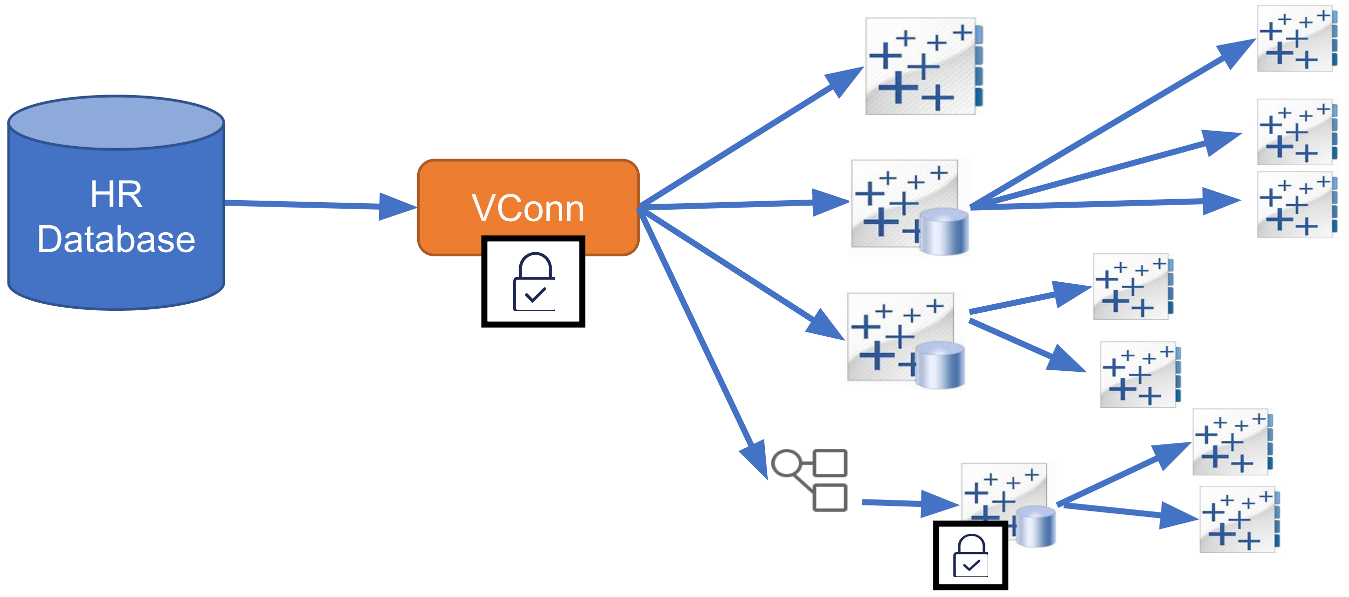
Control Risk



SCALE

What about Centralized Row Level Security?

- New option for implementing RLS in Tableau that requires the Data Management Add-on & version 2021.4 or higher
- We now have Virtual Connections (VCs) which are upstream from data sources and you can apply RLS at the VC layer
- Main benefit is now we have a division of responsibilities: Db admins can create VCs and data policies, while Tableau power users can focus on developing data sources/dashboards



Two Main Types of RLS

- Built-in database RLS methods - LINK
  - Impersonation (MS SQL Server)
  - Kerberos Delegation
  - SAML Delegation (SAP HANA)
  - Initial SQL (Oracle VPD, MS SQL Server, Snowflake)
  - *Strict requirements for these methods to work (live connections, active directory, etc.)
- Entitlements table methods - LINK
  - Recommended and the focus of this talk.. technology vendor agnostic
  - Entitlement = Any unique combination of attributes that the data can be filtered on
  - Two main ways to map your entitlements to users
    - Deepest Granularity Entitlements
    - Sparse Entitlements

What does an entitlement look like?

- Entitlement = Any unique combination of attributes that the data can be filtered on
- Ex. If you filter on Theatre, Region, and Sub-Region, any unique combination of those would be a single Entitlement
- The Entitlement ID, or synthetic key, is an individual Entitlement identifier and the cross-product of the attributes

Entitlements table

Entitlement ID	Theatre	Region	Sub-Region
AMER-USCA-HC	AMER	USCA	HC
AMER-USCA-FS	AMER	USCA	FS
APAC-APAC-JPN	APAC	APAC	JPN

Roles and Users mapping tables

- Very uncommon for Entitlements to be assigned directly to a user
- More commonly, Roles are mapped to Entitlements, and Users are mapped to Roles

Mapping table - Roles to Entitlements

Role ID	Theatre	Region	Role Name	Entitlement ID
RD-AMER	AMER	USCA	Regional Director, Americas	AMER-USCA-HC
RD-AMER	AMER	USCA	Regional Director, Americas	AMER-USCA-FS
RD-APAC	APAC	APAC	Regional Director, APAC	APAC-APAC-JPN
GD	AMER	USCA	Global Director	AMER-USCA-HC
GD	AMER	USCA	Global Director	AMER-USCA-FS
GD	APAC	APAC	Global Director	APAC-APAC-JPN

Mapping table - Users to Roles

Username	User ID	Role ID
rdugger	24490	RD-AMER
jcraycraft	75115	RD-APAC
acortez	32006	GD

+ a b | e a u

Building the Full Entitlements View

- Best practice is to bring the entitlements table & the mapping tables together into a single denormalized View in the db (true for extracts and live connections)

Full Entitlements View

Entitlement ID	Theatre	Region	Sub-Region	Role ID	Username	User ID
AMER-USCA-HC	AMER	USCA	HC	RD-AMER	rdugger	24490
AMER-USCA-FS	AMER	USCA	FS	RD-AMER	rdugger	24490
APAC-APAC-JPN	APAC	APAC	JPN	RD-APAC	jcraycraft	75115
AMER-USCA-HC	AMER	USCA	HC	GD	acortez	32006
AMER-USCA-FS	AMER	USCA	FS	GD	acortez	32006
APAC-APAC-JPN	APAC	APAC	JPN	GD	acortez	32006

Method 1: Deepest Granularity Entitlements

- One row in the table for every entitlement the user has
- Requires less JOIN clauses but more rows in the Entitlements table

Entitlement ID	Theatre	Region	Sub-Region	Role ID	Username	User ID
AMER-USCA-HC	AMER	USCA	HC	RD-AMER	rdugger	24490
AMER-USCA-FS	AMER	USCA	FS	RD-AMER	rdugger	24490
AMER-LAC-BR	AMER	LAC	BR	RD-AMER	rdugger	24490
APAC-APAC-JPN	APAC	APAC	JPN	RD-APAC	jcraycraft	75115
AMER-USCA-HC	AMER	USCA	HC	GD	acortez	32006
AMER-USCA-FS	AMER	USCA	FS	GD	acortez	32006
AMER-LAC-BR	AMER	LAC	BR	GD	acortez	32006
APAC-APAC-JPN	APAC	APAC	JPN	GD	acortez	32006

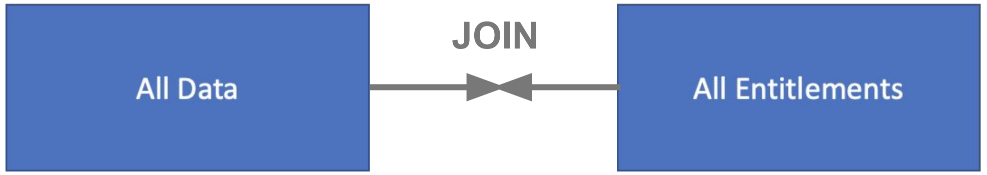
Method 2: Sparse Entitlements

- Entries for every level in a hierarchy, not just entries for the most granular levels of a hierarchy
- NULL values in columns represent having access to ALL of the data at and below that level
- Vastly reduces the number of entitlement rows for users at high levels in a hierarchy

Entitlement ID	Theatre	Region	Sub-Region	Role ID	Username	User ID
ALL	NULL	NULL	NULL	GD	acortez	32006
AMER	AMER	NULL	NULL	RD-AMER	rdugger	24490
AMER-USCA	AMER	USCA	NULL	RD-USCA	psmith	50228
AMER-USCA-HC	AMER	USCA	HC	RD-HC	drector	18819
AMER-USCA-FS	AMER	USCA	FS	RD-FS	acanadien	62377
APAC	APAC	NULL	NULL	RD-APAC	jcraycraft	75115

Best Practice Implementation of RLS in Tableau

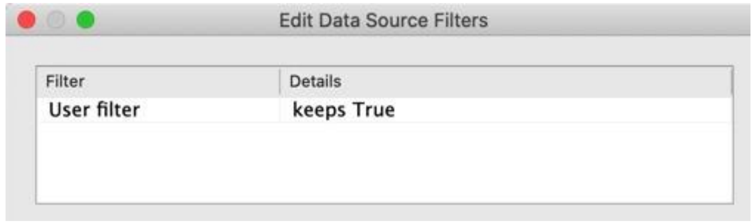
1. Join Data table to Full Entitlements View



2. Create Tableau User calculation

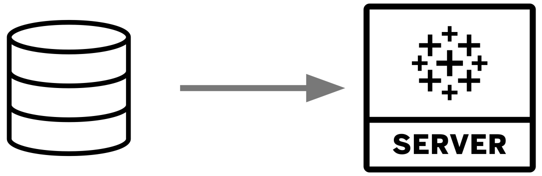
```
USERNAME() = [User]
```

3. Apply calculation as a data source filter

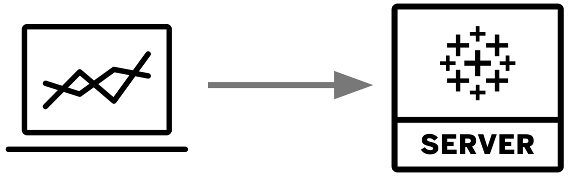

Edit Data Source Filters

Filter	Details
User filter	keeps True

4. Publish data source to Tableau Server



5. Develop with RLS published data sources



RLS Timeline

Pre-2018.3

Legacy RLS methods

Because Tableau had to include all JOINs in a single-table extract, having a fully blown-up entitlements table (deepest granularity) would cause the fact table to blow up too, generating monster size extracts. The Contains RLS Method was an alternative approach, but had many of its own issues.

2018.3

Multi-table extracts released

Multi-table extracts were released in Tableau version 2018.3. The feature allows for storing an extracted data model in Tableau into its separate tables. Now, when customers use deepest granularity entitlements, the extract won't blow up the dataset. Or, they can implement sparse entitlements and aren't required to materialize the join anymore.

2021.4+

Centralized Row Level Security released

Now instead of managing entitlements tables in every data source, they can be managed centrally on Tableau Server/Online in Virtual Connections (VC) and data policies can be enforced in a user-friendly UI. These policies will cascade down to any content that leverages these VCs.

Demo Kit: Deepest Granularity & Sparse Entitlements

RLS Demo Kit

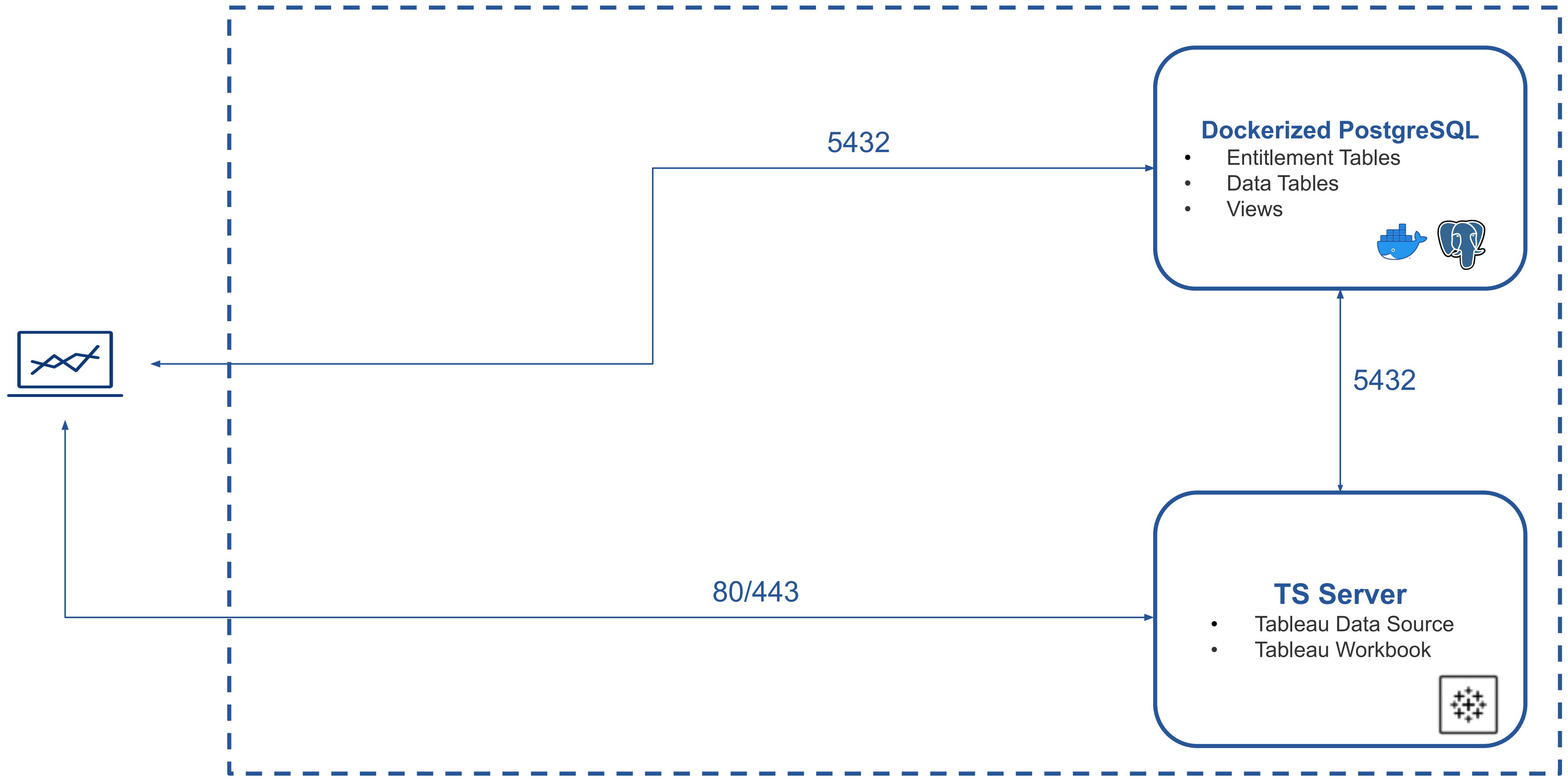
Contents	Description
PostgreSQL running in docker container	● docker-compose ● restore.sh ● postgres data dump
Tableau Workbook	● Deepest Level ● Sparse
Documentation	● Slide deck ● RLS Best Practices Whitepaper

https://github.com/jrunrun/tableau-row-level-security-demo

RLS Demo Kit - How to Install and Run

1. Assuming you have git and docker installed, execute the following commands from the command line:
   - git clone https://github.com/jrunrun/tableau-row-level-security-demo
   - docker-compose up

Network Architecture for RLS Demo Kit



Full Implementation Steps of RLS

Steps	Deepest Level Entitlements	Sparse Entitlements
1. Join Data table to Full Entitlements View	One join ● Entitlement ID (data) = Entitlement ID (entitlements View) Multiple joins ● Theatre ID (data) = Theatre ID (entitlements View) ● Region ID (data) = Region ID (entitlements View) ● Sub-Region ID (data) Sub-Region ID (entitlements View)	One join ● 1 = 1
2. Create Tableau User calculation	One calculation ● Username() = [Username]	Two calculations ● Username() = [Username] ● Condition calc
3. Apply calculation as a data source filter	One filter ● Username calc = True	Two filters ● Username calc = True ● Condition calc = True
4. Publish data source to Tableau Server	Same	Same
5. Develop with RLS published data sources	Same	Same

Contrasting Deepest Level and Sparse

- Regardless of the model used to represent the entitlements, it is advisable to join all entitlements/mapping tables together into a single denormalized entitlements View. While at first this will cause a “blowup” (highly duplicative) version of the entitlements, the entitlements should be filtered down to just those belonging to a particular user before it gets joined into the data.

Deepest Level Entitlements	Sparse Entitlements
Entitlements are defined fully for every column.	Entitlements are defined for every level of hierarchy, with NULL used to represent an “all” state.
Simpler approach: There is one row in the mapping table for every possible entitlement the user has.	There is a single row in the mapping table for a particular level in the entitlement hierarchy, which vastly reduces the number of entitlement rows for users at high levels in a hierarchy.
This model may require fewer join clauses.	This model requires more complex joins and filters.

Q&A

tableau®