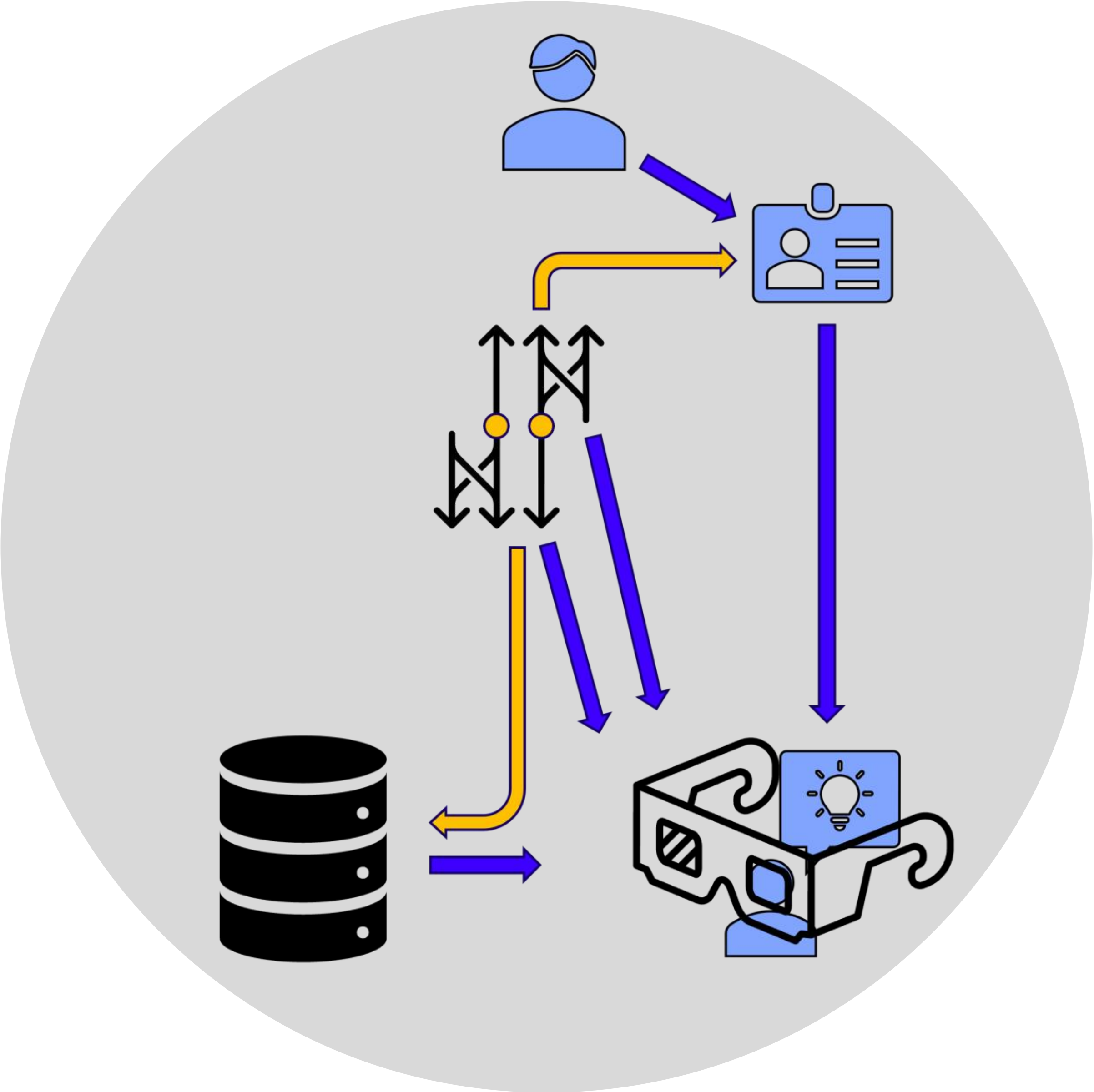
Handling complex Row- and Column-Level-Security at scale in Databricks



Enrico Schnepel

DataGrillen, 2024-05-16

Enrico Schnepel

- Senior Expert Data Engineering
- Business Intelligence
- SoftwareOne | Leipzig

- 3 decades with one question:
    - How to process data *better* ?

- Databricks (SparkQL / PySpark)
- SQL Server
- SSAS Tabular Model

softwareone

Handling complex Row- and Column-Level-Security at scale in Databricks

Introduction - 1

What is … ?

Proof of Concept

Requirements

Concept - 2

Evolution

- Concept 0.9
- Concept 1.0

Concept 2.0

- User & Data
- Operator Logic

Processing - 3

User data

Fact data

Mapping

View

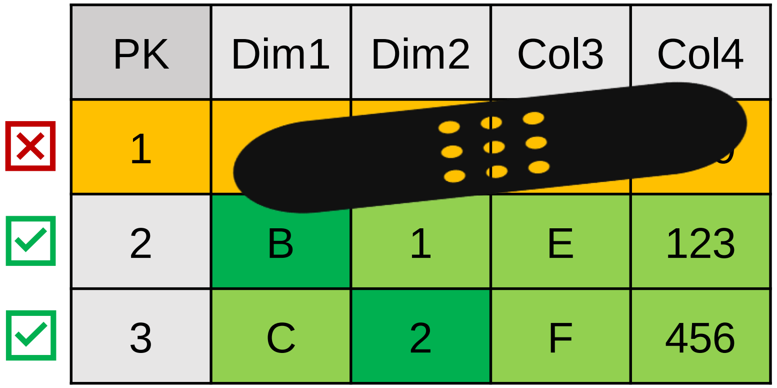
Conclusion

Q&A

1.1 – What is … ?

Row Level Security (RLS)

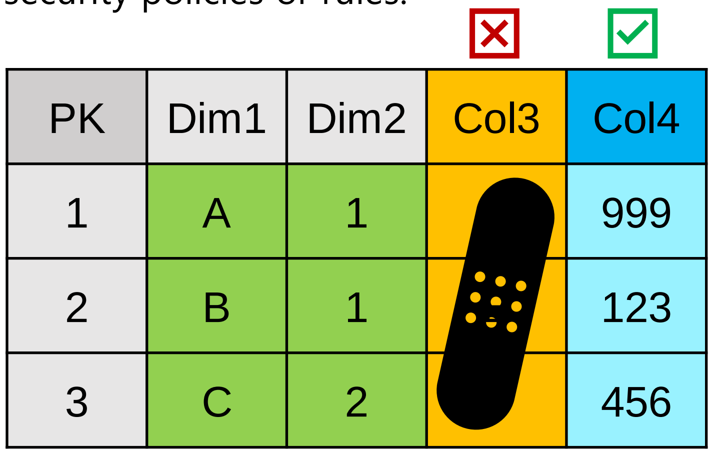
is a data protection feature that restricts user access to specific rows of data in a table. It allows you to control which users can view (or edit) specific rows of data based on predefined security policies or rules.



	PK	Dim1	Dim2	Col3	Col4
☒	1				
☑	2	B	1	E	123
☑	3	C	2	F	456

Column Level Security (CLS)

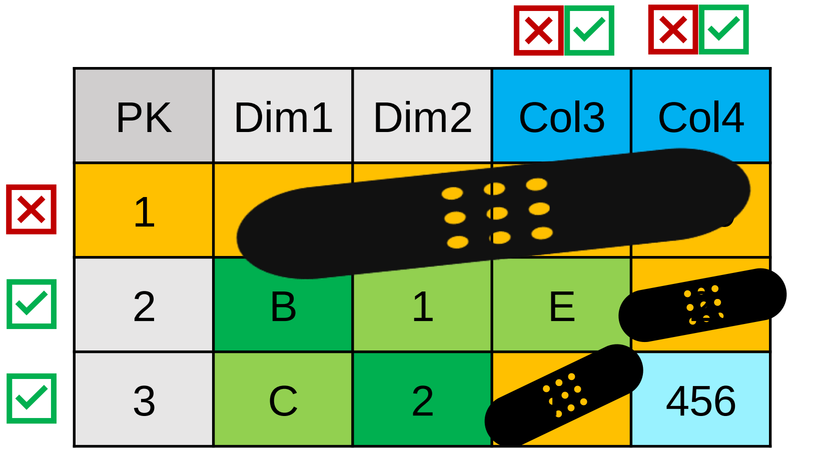
is a data protection feature that restricts user access to specific columns of data in a database table. It allows you to control which users can view (or edit) specific columns of data based on predefined security policies or rules.



			☒	☑
PK	Dim1	Dim2	Col3	Col4
1	A	1		999
2	B	1		123
3	C	2		456

Row & Column Level Security (RCLS)

Combines both approaches and adds redaction to cells where RLS would allow the visibility of the row, but CLS prohibits showing column data for that row.



				☒☑	☒☑
	PK	Dim1	Dim2	Col3	Col4
☒	1				
☑	2	B	1	E	
☑	3	C	2		456

1.1 – What is … ?

OBJECT LEVEL SECURITY (OLS)

Row Level Secu

is a data p
user acces
database
which use
rows of d
policies o rules.

ecurity (RCLS)

nes and adds
RLS would allow
but CLS prohibits
a for that row.

PK	Dim1	Dim2	Col3	Col4

1

1

E

C

2

45

36

1.2 – Proof of Concept

- OnPrem environment (SQL-Server) planned to be decomissioned

- Migration of processes towards a cloud environment
  - Databricks as processing platform
  - Azure DeltaLake as storage

- How to handle Row- and Column Level Security (RCLS) in Databricks?

https://www.pinterest.com/pin/690598924084602624/ [2024-03-26]

1.3 – Requirements

Configuration

- Multiple fact tables
- Multiple business domains
- Different access groups

- Optional:
  - Sensitive data

Process

- Complex RLS rules
- m:n mapping of dimension values

- Optional:
  - CLS depends on RLS
  - Exclusions

User

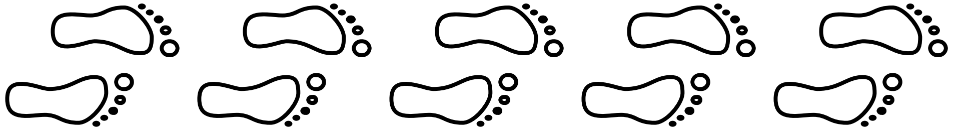
- My data – now !
- Minimal RLS overhead for end user reports
- Minimal lag for data processing

1.3 – Requirements – Developer

The argument

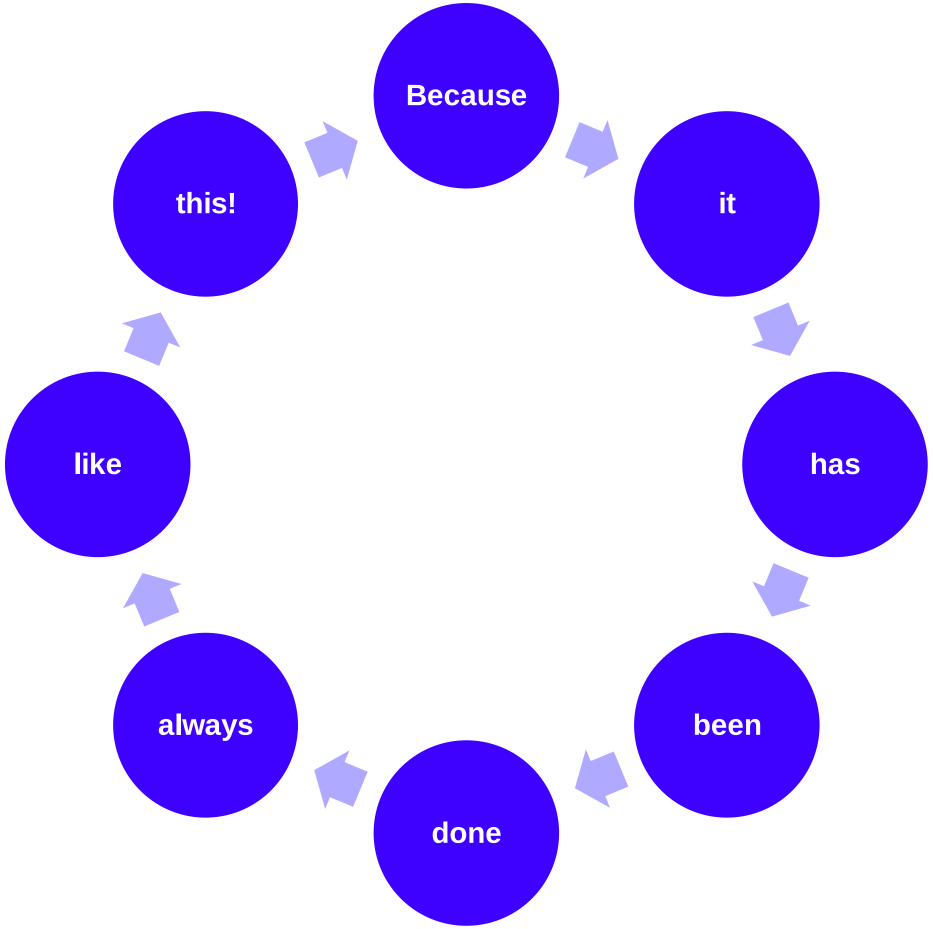
“Because it has been done always like this!”

is a strong one …



But if you repeat too often, errors will appear…

R Bicaise t ueaue üt ?as been done always’a lil,YE Ti$!



1.3 – Requirements – Developer

Rethink

Turn around and look. The argument “**Because it has been done always like this**” is a strong one … but not a reason to do it the same way again. May be the environment has changed.

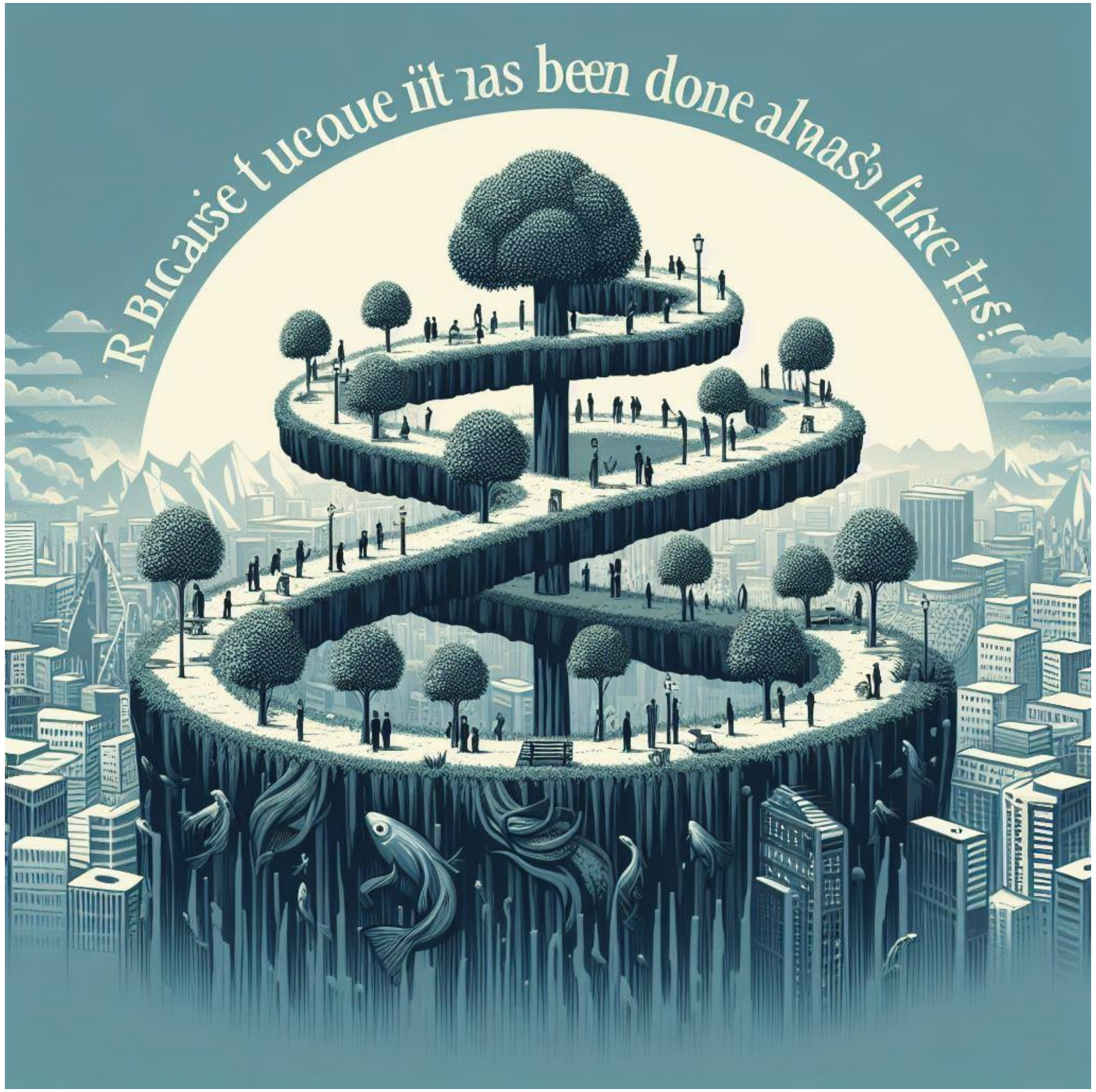
➥Abstract!

Abstract

Abstract the next big question from the challenges you see.

➥ Rethink!

CONSTANTLY EVOLVE!



Generated with AI using the “Rethink” text content on February 8, 2024 at 1:18 PM

1.3 – Requirements – Developer

Go the second-intuitive way

Is there a way next to the obvious one? Let’s try the first steps!

O there is an elevator which can be used instead of the staircase…

Sometimes you must build or even invent something first
before you can see the advantages.

Have fun!

Generated with AI using the prompt “Please create a picture of two people. One is pointing in the obvious direction towards a staircase, the other towards a construction place where an elevator is currently installed.” on February 8, 2024 at 1:42 PM

Handling complex Row- and Column-Level-Security at scale in Databricks

Introduction - 1

What is … ?

Proof of Concept

Requirements

Concept - 2

Evolution

- Concept 0.9
- Concept 1.0

Concept 2.0

- User & Data
- Operator Logic

Processing - 3

User data

Fact data

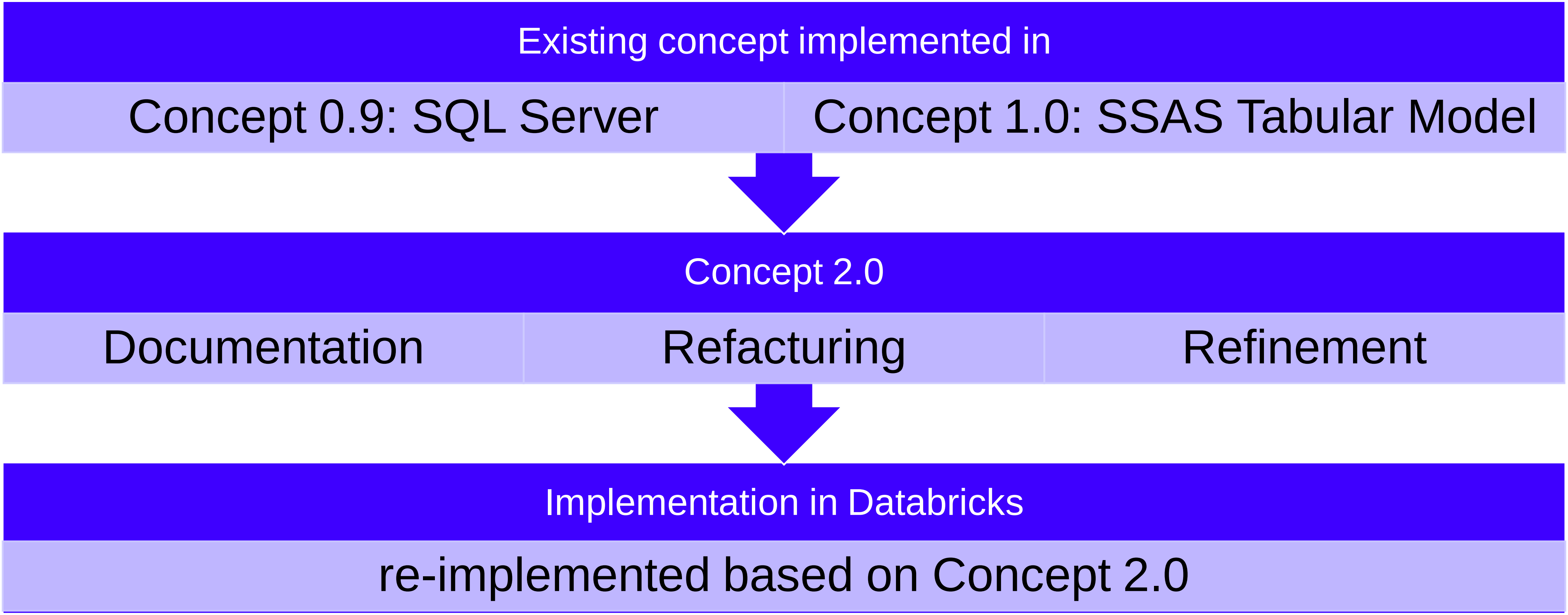
Mapping

View

Conclusion

Q&A

2.1 – Proof of Concept – RLS in Databricks



Existing concept implemented in

Concept 0.9: SQL Server | Concept 1.0: SSAS Tabular Model

Concept 2.0

Documentation | Refactoring | Refinement

Implementation in Databricks

re-implemented based on Concept 2.0

2.2 – Concept 0.9 – The Classical Approach (1)

SQL Server

- One business domain (HR)
- Sensitive data
- Two fact tables
- Different access groups
- Complex RLS rules
  (AND and OR operators)
- 2 CLS config flags, later 3
- CLS depends on RLS

- Reports just call a …

  Table-valued-function for handling RLS
  and get CLS metadata as result

SSAS Tabular Model – 1st try

- Reimplementing the TV-function
  as complex DAX expression
- directly in the TM
- using the existing rule table
- one fact table
- 3 out of 5 dimensions for RLS
- 1 CLS config flags

- Way to slow during runtime

2.2 – Concept 0.9 – The Classical Approach (2)

RLS Rule sets written

- as a conditional expression:

User X: Dim1 = A **and** Dim2 = 5

User Y: Dim1 = B **or** Dim2 = 2

- and as a table …

```
CREATE OR REPLACE VIEW v_fact AS
SELECT
  fact.*
FROM fact
WHERE EXISTS (
    SELECT *
    FROM rls
    WHERE rls.AccountName = user_account_name()
      AND (fact.Dim1 = rls.Dim1 OR rls.Dim1 IS NULL)
      AND (fact.Dim2 = rls.Dim2 OR rls.Dim2 IS NULL)
  )
```

User	Rule	Dim1	Dim2		Fact_PK	Dim1	Dim2	Amount	Count
X	1	A	5	≠		A	1	100	8
Y	2	B	*NULL*	=	2	B	1	200	2
Y	3	*NULL*	2	=	3	C	2	500	3

2.3 – Concept 1.0 – Pre-generated mapping of RLS rules (3)

SSAS Tabular Model – 2nd try

SQL Server

- Generation of mapping tables between user RLS rules and fact dimensions
- 2 fact tables, later 4
- 4 dimensions for RLS, later 6
- 1 CLS config flags, later 2

SSAS Tabular Model

- Loading of mapping tables
- RLS: Filter for the username
- Using native TM functionality
    - Much faster

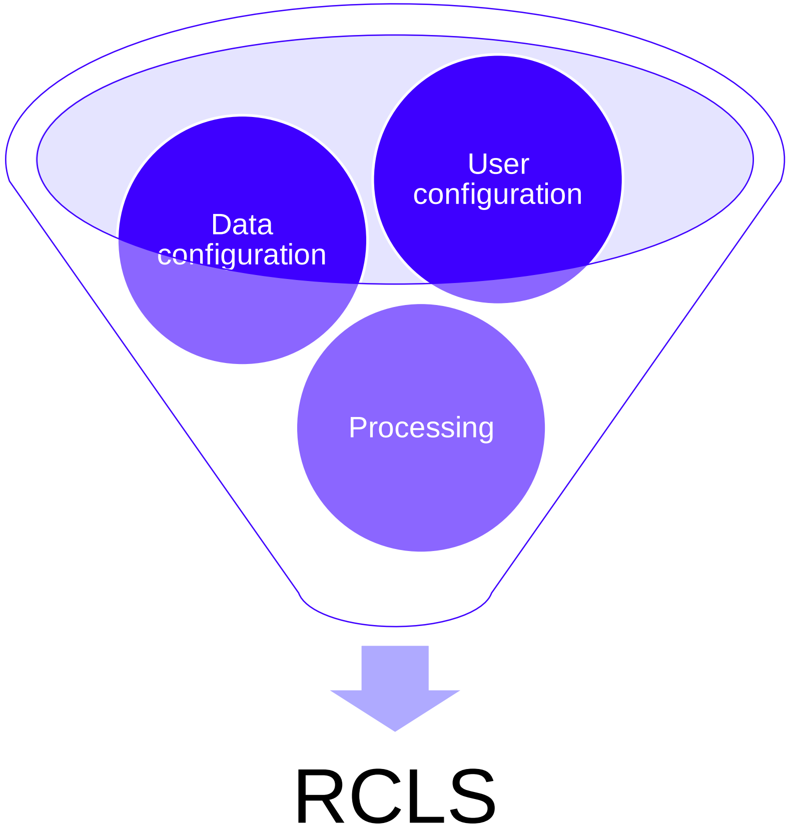
2.3 – Concept 1.0 – Pre-generated mapping of RLS rules (1)

Idea for Pre-Processing RCLS

- Pre-process relation between user account and fact data
  - Avoid time-consuming re-evaluation of RLS rules for each report access
- Instead of column-wise mapping of RLS rules to RLS conditions
  - Transpose fact table and RLS rule columns (similar on both sides)
  - Join and Aggregate
  - Evaluate and Filter
- Easy to handle (for the database)

Concept 1.1:

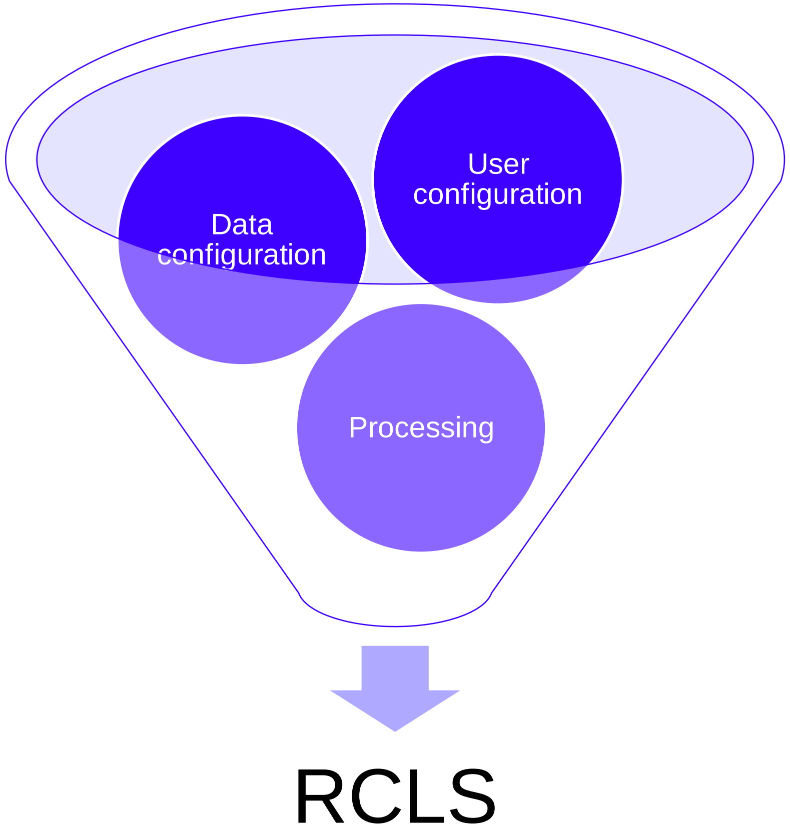
- Support for supervisor mapping



2.3 – Concept 1.0 – Pre-generated mapping of RLS rules (2)

Tabular Model

- Fact table joined with 2 background mapping tables
  - User2Access mapping table
    - Filtered using the current user account name
  - Access2Data mapping table
  - Fact Data table
    - Filtered with an inner join
- Fast single-column joins between table pairs



2.3 – Concept 1.0 – Operator logic – Transpose dimensions

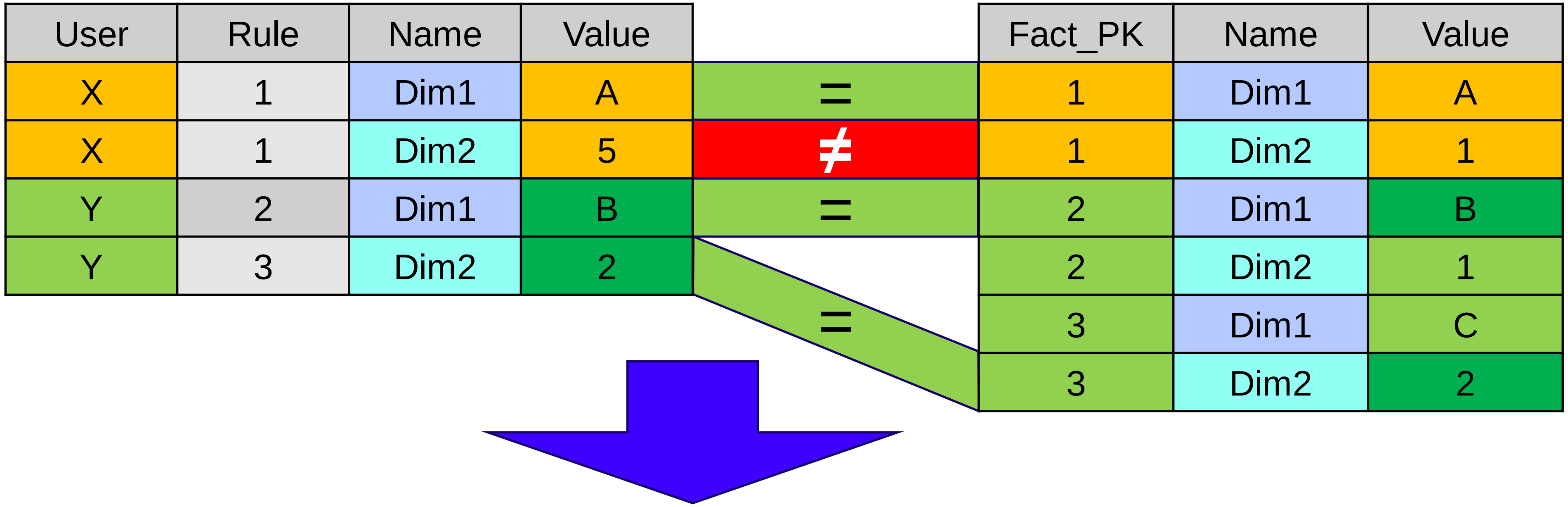
User	Rule	Dim1	Dim2
X	1	A	5
Y	2	B	*NULL*
Y	3	*NULL*	2

User	Rule	Name	Value
X	1	Dim1	A
X	1	Dim2	5
Y	2	Dim1	B
Y	3	Dim2	2

Fact_PK	Dim1	Dim2	Amount	Count
1	A	1	100	8
2	B	1	200	2
3	C	2	500	3

Fact_PK	Name	Value
1	Dim1	A
1	Dim2	1
2	Dim1	B
2	Dim2	1
3	Dim1	C
3	Dim2	2

2.3 – Concept 1.0 – Operator logic – Join, Aggregate and Filter



User	Rule	Name	Value
X	1	Dim1	A
X	1	Dim2	5
Y	2	Dim1	B
Y	3	Dim2	2

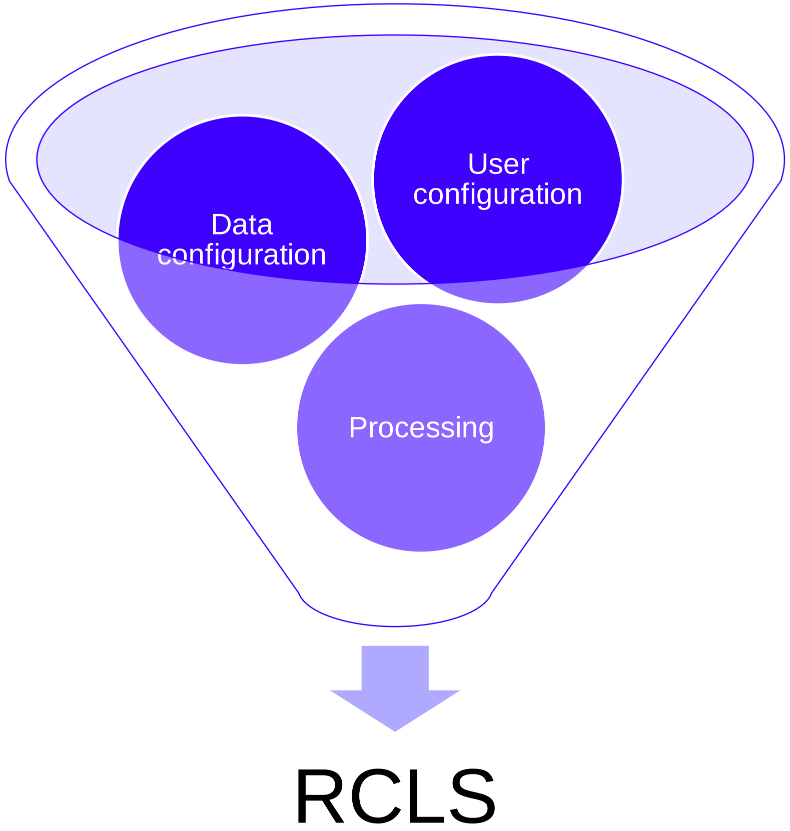
Fact_PK	Name	Value
1	Dim1	A
1	Dim2	1
2	Dim1	B
2	Dim2	1
3	Dim1	C
3	Dim2	2

User	Rule	Expected Count	Fact_PK	Matched dimension values	Access granted?
X	1	2	1	1	**No**
Y	2	1	2	1	**Yes**
Y	3	1	3	1	**Yes**

2.4 – Concept 2.0

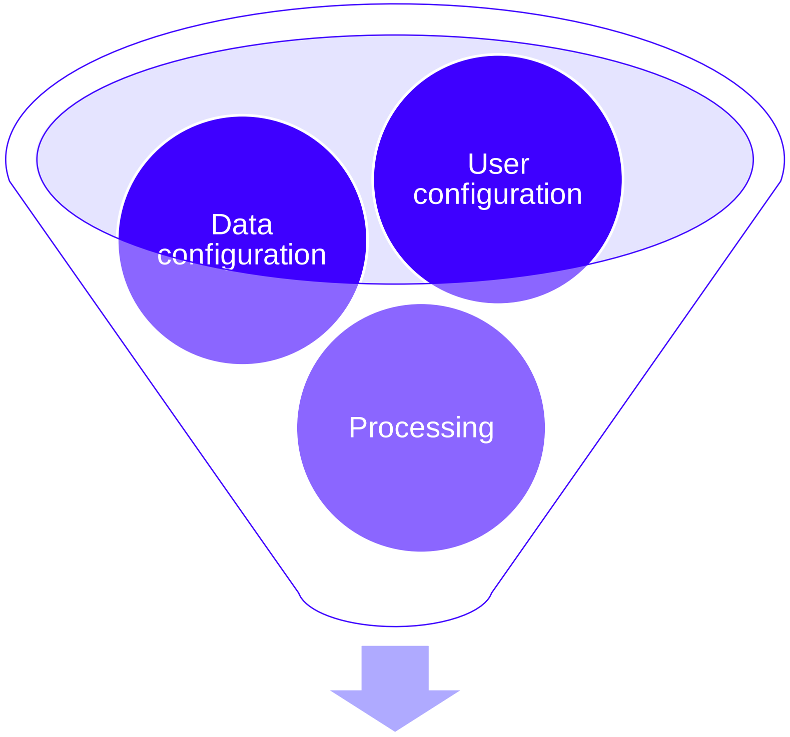
Main differences to Concept 1.1

- Documented
- Configuration-driven processing of RLS rules
- MapReduce concept for processing logical operators
  - Support for NOT / EXCEPT and IN operator
- Generalized support for m:n mapping tables
- intermediate mapping tables are clustered by the logical attribute name
- Generated View is used in reports or by Power Users (equivalent to Tabular Model for Concept 1.0 / 1.1)



2.4 – Concept 2.0 – User configuration

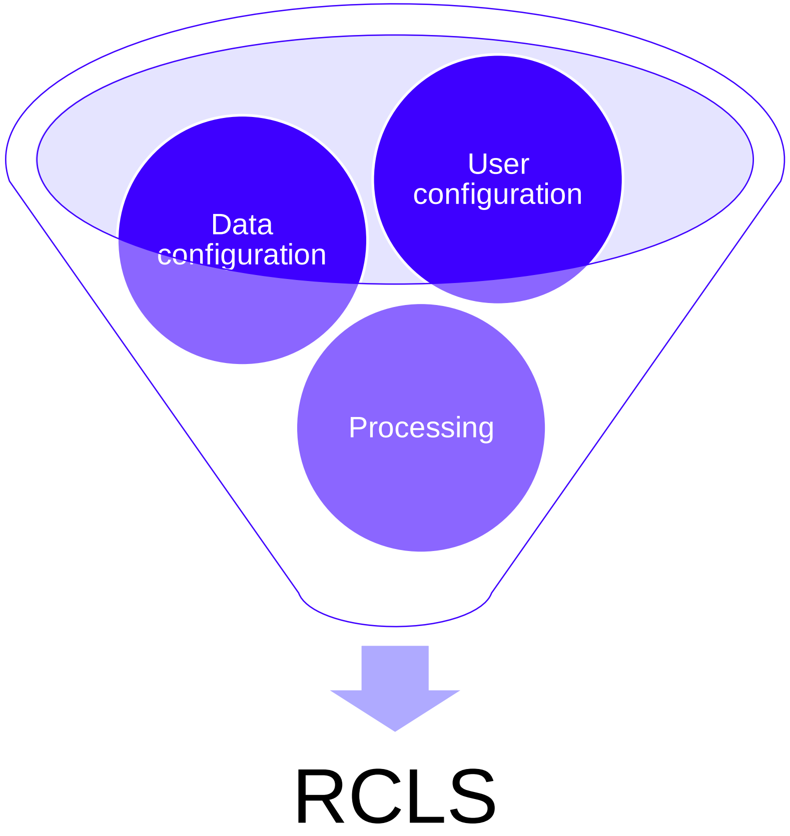
- Assignments to business domains
- RLS rules without technical details (like fact column names)
- Examples:
  - Sales data for country Germany
  - During absence same access as supervisor on sales data
  - Aggregated finance data worldwide down to country level



RCLS

2.4 – Concept 2.0 – Data configuration

- Mapping fact tables to business domains
- Mapping physical fact table column names
  to logical RLS column names
  - physical “Sales”.“Country_Name“ = logical “country”
- Generalized dynamic m:n or hierarchy mapping
  - supervisor hierarchy
  - “worldwide” = [list, of, all, country, names]



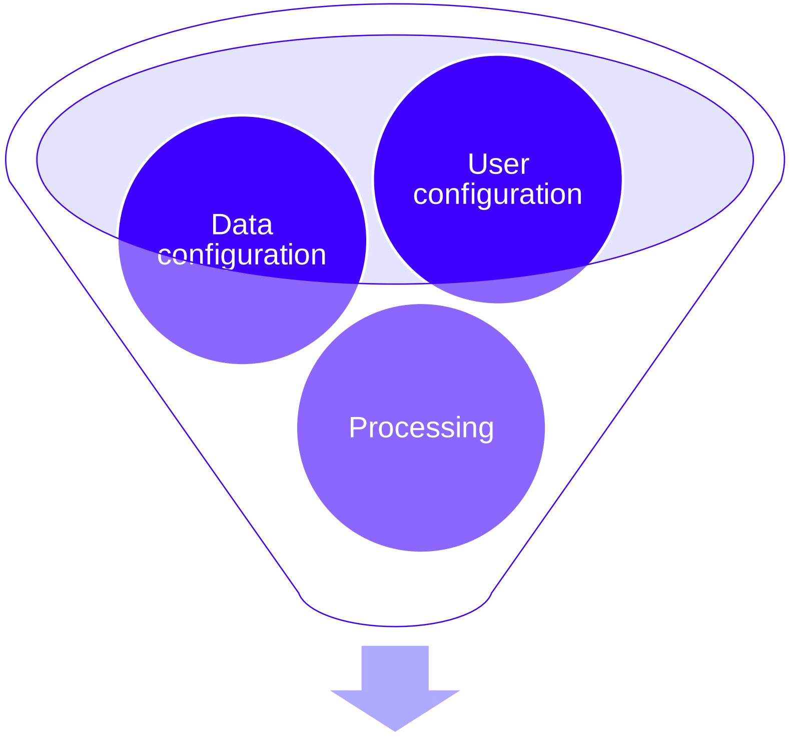
2.4 – Concept 2.0 – User and Data configuration

User configuration

- Assignments to **business domains**
- RLS rules **without** technical details (like fact column names)
  - **Sales** data for **country** Germany
  - During absence: same access as **supervisor** on **sales** data
  - Aggregated **finance** data **worldwide** down to **country** level

Data configuration

- Mapping fact tables to **business domains**
- Mapping **fact column names** to **logical column names**
  - physical “**Sales**”.“**Country_Name**“ = “**country**”
- Dynamic m:n or hierarchy mapping
  - **supervisor** hierarchy
  - “**worldwide**” = [list, of, all, country, names, e.g., Germany]



RCLS

2.4 – Concept 2.0 – Operator logic – Support for EXCEPT and IN

Logical operation	Expected matching column		Matching column		Access granted?
	names	values	names	values	
AND	Exact 2	Exact 2	2	2	Yes
			2	1	No
			1	1	No
OR	Min 1	Min 1	2	2	Yes
			1 or 2	1	Yes
			1	0	No
NOT / EXCEPT	Exact 1	Exact 0	1	1	No
			1	0	Yes
IN	Not handled specially				

Handling complex Row- and Column-Level-Security at scale in Databricks

Introduction - 1

What is … ?

Proof of Concept

Requirements

Concept - 2

Evolution

- Concept 0.9
- Concept 1.0

Concept 2.0

- User & Data
- Operator Logic

Processing - 3

User data

Fact data

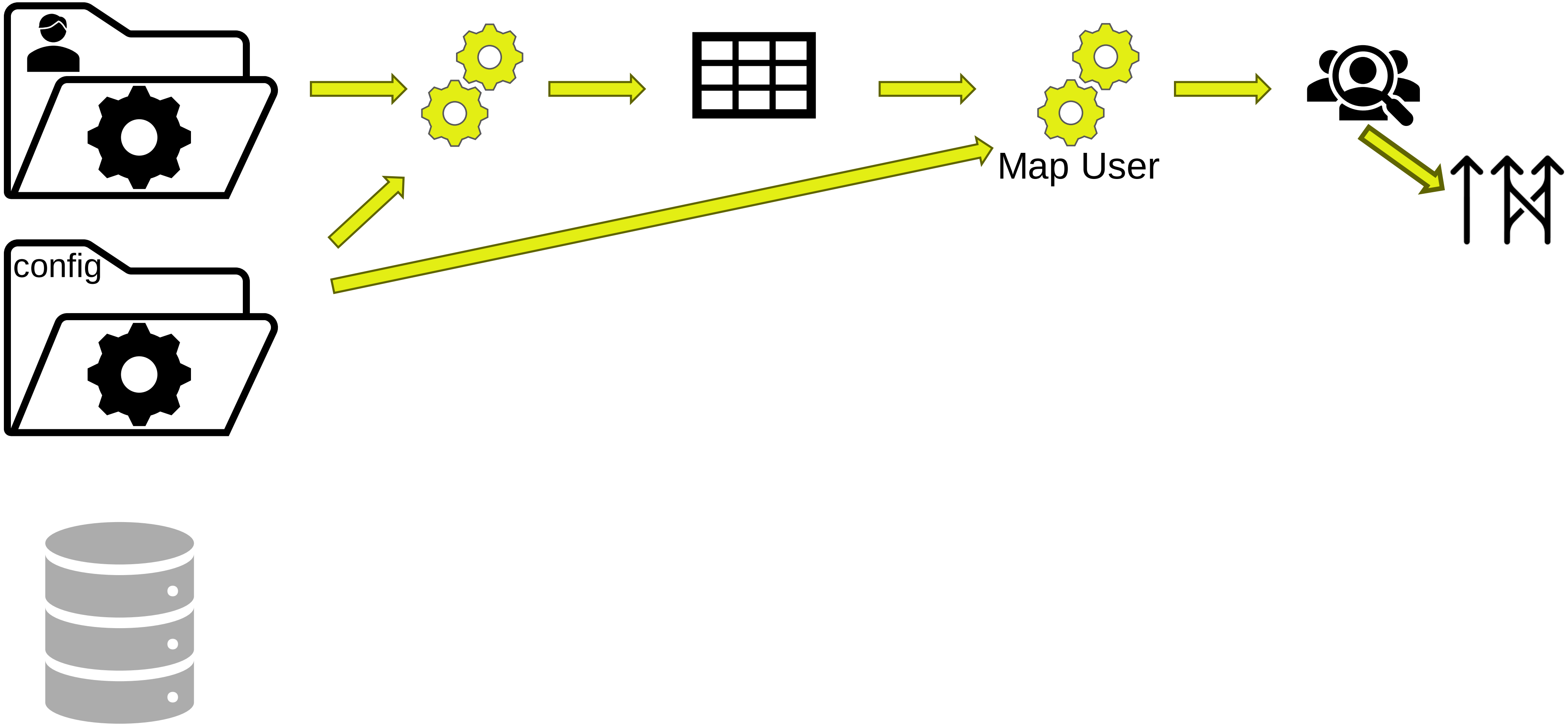
Mapping

View

Conclusion

Q&A

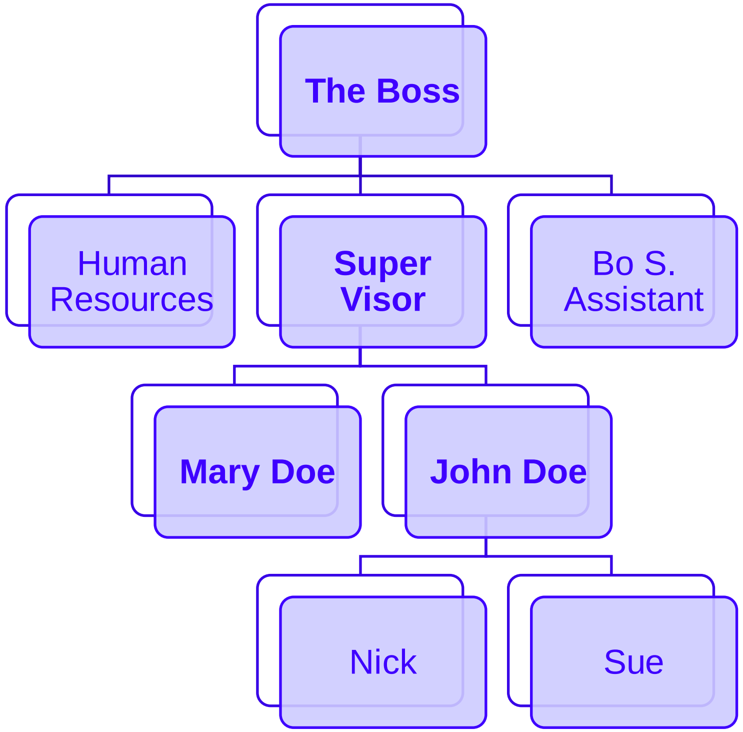
3.1 – The Process Workflow (User data)



3.1 – User – dynamic mapping (1)

Generate a „**dynamic mapping**“ table:

- E.g. a Supervisor Hierarchy



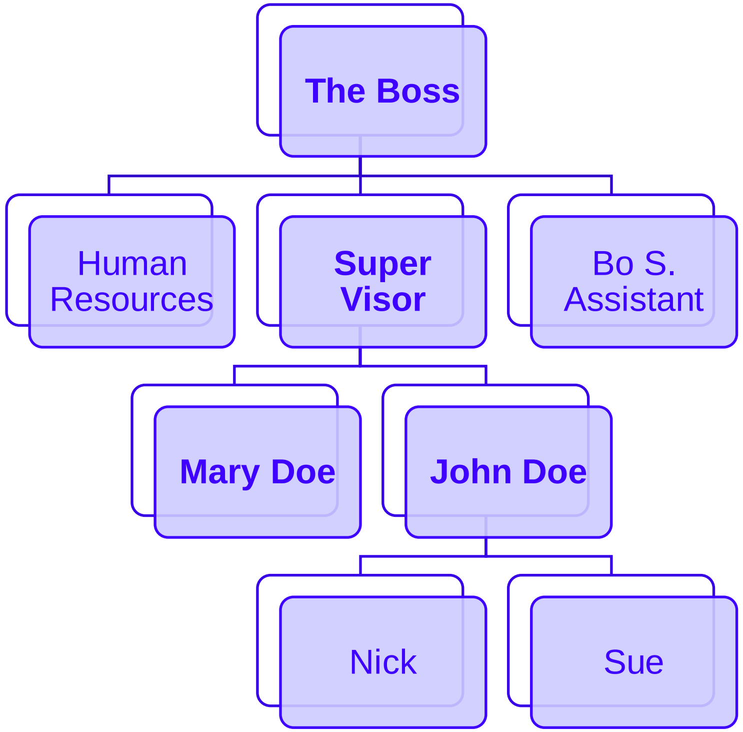
User Name	Direct Reports (DR)	Indirect Reports
The Boss	Super Visor, Bo S. Assistant, Human Resources	John Doe, Mary Doe, Nick, Sue
Super Visor	John Doe, Mary Doe	Nick, Sue
John Doe	Nick, Sue	
No DR:	*Bo S. Assistant, Human Resources, Mary Doe, Nick, Sue*	

User Name	Direct Reports (DR)	SV of Indirect Reports
The Boss	Super Visor, Bo S. Assistant, Human Resources	DR of “Super Visor” DR of “John Doe”
Super Visor	John Doe, Mary Doe	DR of “John Doe”
John Doe	Nick, Sue	
No DR:	*Bo S. Assistant, Human Resources, Mary Doe, Nick, Sue*	

3.1 – User – dynamic mapping (2)

Generate a „**dynamic mapping**“ table:

- e.g. a Supervisor Hierarchy, Regional Hierarchy



Attribute	Value	MappedValue
Hierarchy.Self	John Doe	John Doe
Hierarchy.Self	Mary Doe	Mary Doe
Hierarchy.Self	Super Visor	Super Visor
Hierarchy.Self	The Boss	The Boss
Hierarchy.Self	…	…
Hierarchy.DirectReports	John Doe	Nick
		Sue
Hierarchy.DirectReports	Super Visor	John Doe
		Mary Doe
Hierarchy.DirectReports	The Boss	Super Visor
		Human Resources
		Bo S. Assistant
Hierarchy.**In**DirectReports	John Doe	Nick
		Sue
Hierarchy.**In**DirectReports	Super Visor	John Doe
		Mary Doe

3.1 – User – static / dynamic setup + AD-Group configuration

User Name	Department
John Doe	Sales
Mary Doe	BackOffice
Super Visor	BackOffice

User Name	Attribute	Value
John Doe	Country	DE
John Doe	Country	FR
John Doe	CostCenter	Software
Mary Doe	CostCenter	Finance

Attribute	Value
Hierarchy.Self	John I
Hierarchy.Self	Mary
Hierarchy.Self	Sup
Hierarchy.Self	Th
Hierarchy.Self	..
Hierarchy.DirectReports	J
Hierarchy.DirectReports	Sup
Hierarchy.DirectReports	The B
Hierarchy.**In**DirectReports	Joh
Hierarchy.**In**DirectReports	S

User Name	AD-Group
John Doe	Finance
John Doe	Sales
Mary Doe	Global
Super Visor	Global

AD-Group	Bus-Domain
Finance	Finance
Sales	Sales
Global	Finance
Global	Sales

3.1 – User – access attribute mapping

Generate a „**user access attribute mapping**“ table:

- which can be used for providing access
- to any user

- Static user setup
  - Explicit Country and CostCenter access
- Dynamic user setup
  - Department
- Dynamic user mapping
  - Hierarchy.Self
  - Hierarchy.DirectReports
  - Hierarchy.InDirectReports

User Name	Attribute	Value
John Doe	Country	DE
John Doe	Country	FR
John Doe	CostCenter	Software
John Doe	Department	Sales
John Doe	Hierarchy.Self	John Doe
John Doe	Hierarchy.DirectReports	John Doe
Mary Doe	CostCenter	Finance
Mary Doe	Department	BackOffice
Mary Doe	Hierarchy.Self	Mary Doe
Super Visor	Department	BackOffice
Super Visor	Hierarchy.Self	Super Visor
Super Visor	Hierarchy.DirectReports	Super Visor
Super Visor	Hierarchy.**In**DirectReports	John Doe
The Boss	Hierarchy.Self	The Boss
The Boss	Hierarchy.DirectReports	The Boss
The Boss	Hierarchy.**In**DirectReports	Super Visor
The Boss	Hierarchy.**In**DirectReports	John Doe

3.1 – User – rule configuration

AD-Group – Object-Level Access to:

- Bracket for multiple rules, merged with an OR operator
- SSAS (Cube, Tabular Model), Reports (SSRS), PowerBI (App)

Rule-Name:

- Bracket for AND Operator spanning multiple attributes
- More precise rules (more columns) overrule lover column counts

Attribute:

- IN-Operator for multiple values for the same user and attribute name
- Referencing the logical attribute for the static or dynamic configuration

AD-Group	Rule-Name	Attribute
Finance	**CountryAndCC**	**Country**
Finance	**CountryAndCC**	**CostCenter**
Finance	Country	Country
Finance	CostCenter	CostCenter
Sales	Country	Country
Global	Country	Country
Global	CostCenter	CostCenter
Sales	Hierarchy.Self	Hierarchy.Self
Sales	Hierarchy.DirectReports	Hierarchy.DirectReports
Global	Hierarchy.Self	Hierarchy.Self
Global	Hierarchy.DirectReports	Hierarchy.DirectReports
Global	Hierarchy.InDirectReports	Hierarchy.InDirectReports

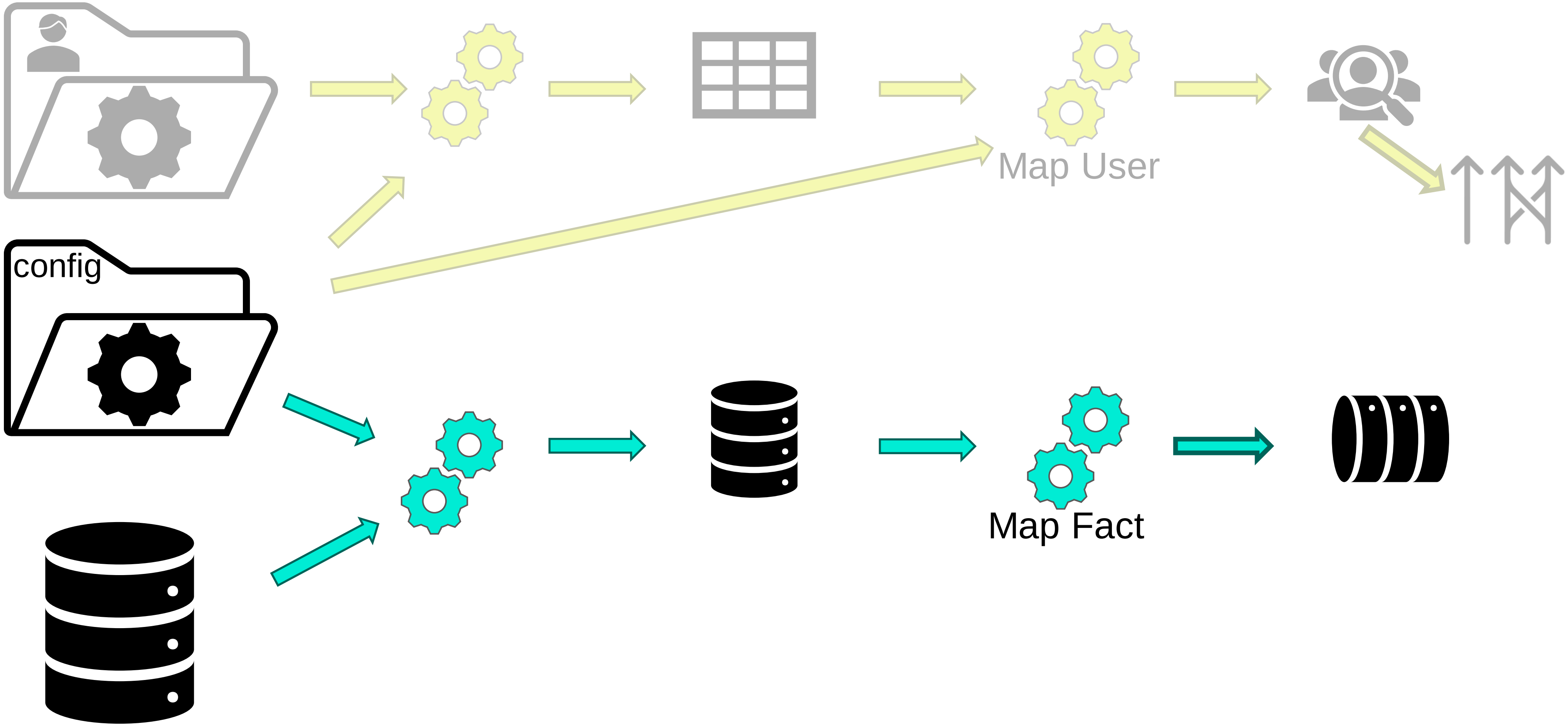
3.1 – User – access mapping (1)

User Name	AD-Group	Rule-Name	Attribute	Value	Bus-Domain	Comment
John Doe	Finance	CountryAndCC	Country	DE	Finance	More precise than Rows 4 to 6
John Doe	Finance	CountryAndCC	Country	FR	Finance	
John Doe	Finance	CountryAndCC	CostCenter	Software	Finance	
~~John Doe~~	~~Finance~~	~~Country~~	~~Country~~	~~DE~~	~~Finance~~	overruled
~~John Doe~~	~~Finance~~	~~Country~~	~~Country~~	~~FR~~	~~Finance~~	overruled
~~John Doe~~	~~Finance~~	~~CostCenter~~	~~CostCenter~~	~~Software~~	~~Finance~~	overruled
John Doe	Sales	Country	Country	DE	Sales	
John Doe	Sales	Country	Country	FR	Sales	
John Doe	Sales	Hierarchy.Self	Hierarchy.Self	John Doe	Sales	
John Doe	Sales	Hierarchy. DirectReports	Hierarchy. DirectReports	John Doe	Sales	
~~John Doe~~			~~Department~~	~~Sales~~		No Rule with Department

3.1 – User – access mapping (2)

User Name	AD-Group	Rule-Name	Attribute	Value	Bus-Domain	Comment
	~~Global~~	~~Country~~	~~Country~~			Not matched by user
Mary Doe	Global	CostCenter	CostCenter	Finance	Finance	
Mary Doe	Global	CostCenter	CostCenter	Finance	Sales	
Mary Doe	Global	Hierarchy	Self	Mary Doe	Sales	
Super Visor	Global	Hierarchy.Self	Hierarchy.Self	Super Visor	Sales	
Super Visor	Global	Hierarchy.DirectReports	Hierarchy.DirectReports	Super Visor	Sales	
Super Visor	Global	Hierarchy.InDirectReports	Hierarchy.InDirectReports	John Doe	Sales	
The Boss	Global	Hierarchy.Self	Hierarchy.Self	The Boss	Sales	
The Boss	Global	Hierarchy.DirectReports	Hierarchy.DirectReports	The Boss	Sales	
The Boss	Global	Hierarchy.InDirectReports	Hierarchy.InDirectReports	Super Visor	Sales	
The Boss	Global	Hierarchy.InDirectReports	Hierarchy.InDirectReports	John Doe	Sales	

3.2 – The Process Workflow (Fact data)



3.2 – Fact data

Sales Fact table

PK	CountryEN	CostCenter	FK_Manufact	FK_Product	Status	Count	Price	Amount	DimAccessKey
1	**DE**	**Software**	**2**	5	Offered	2	5,67	11,34	**43AD56**
2	**DE**	**Software**	**7**	3	Invoiced	1	6,78	13,56	**9874FE**
3	**FR**	**Cloud**	**9**	4	Offered	3	9,99	29,97	**7E65AC**
4	**NL**	**Cloud**	**9**	6	Invoiced	5	1,00	5	**237561**
5	**NL**	**Cloud**	**9**	7	Offered	4	2,34	9,36	**237561**

DISTINCT

Sales Fact DimensionAccess Table

DimAccessKey	CountryEN	CostCenter	FK_Manufact
43AD56	DE	Software	2
9874FE	DE	Software	7
7E65AC	FR	Cloud	9
237561	NL	Cloud	9

3.2 – Fact data

Sales Fact table

PK	CountryEN	CostCenter	FK_Manufact	FK_Prod
1	DE	Software	2	5
2	DE	Software	7	
3	FR	Cloud	9	4
4	NL	Cloud	9	6
5	NL	Cloud	9	

DISTINCT

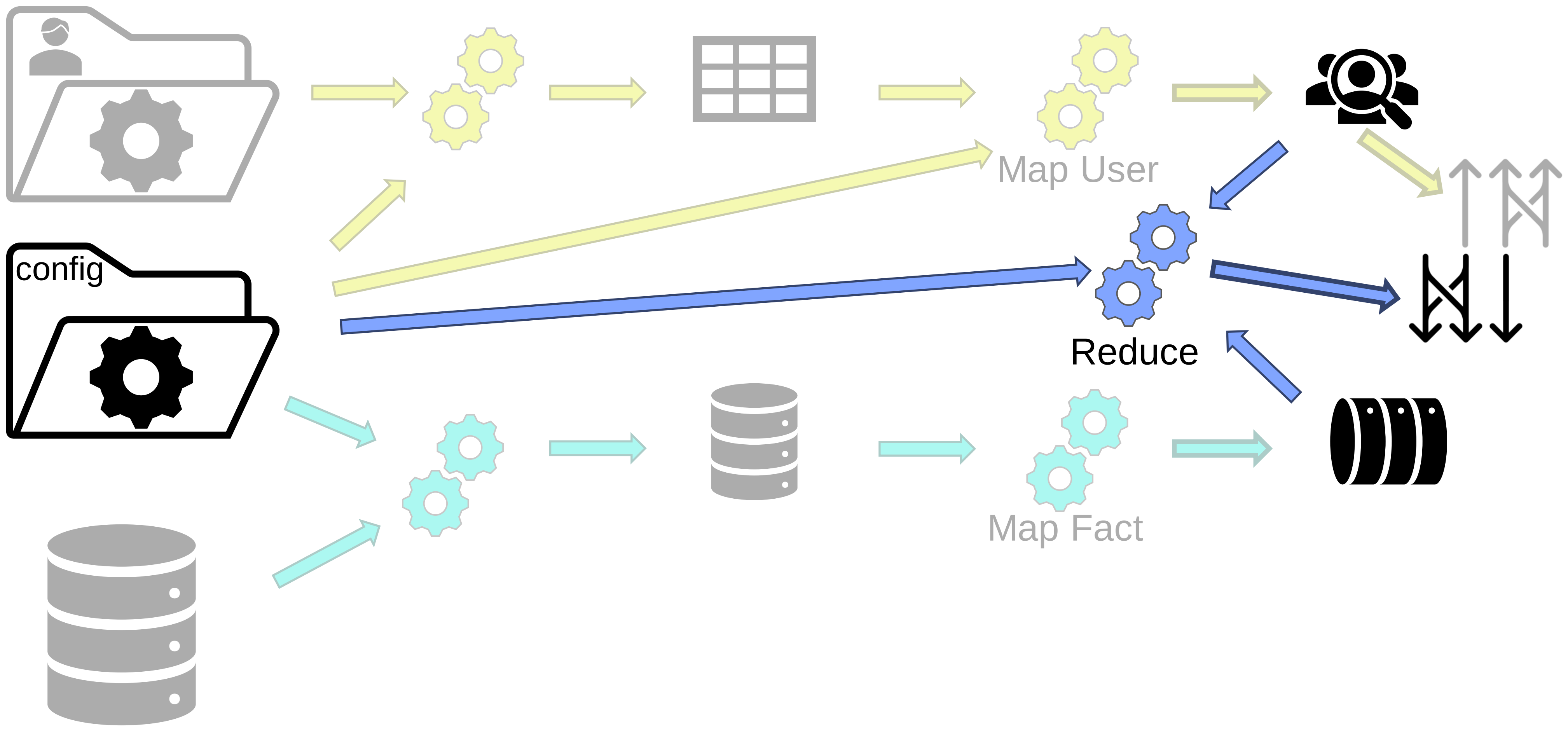
Sales Fact Dimension**Access** Table

DimAccessKey	CountryEN	CostCenter	FK_Manufact
43AD56	GE	Software	2
9874FE	GE	Software	7
7E65AC	FR	Cloud	9
237561	NL	Cloud	9

Sales Fact Dimension**Mapping** Table

DimAccessKey	ColumnName	ColumnValue
43AD56	CountryEN	GE
43AD56	CostCenter	Software
43AD56	FK_Manufact	2
9874FE	CountryEN	GE
9874FE	CostCenter	Software
9874FE	FK_Manufact	7
7E65AC	CountryEN	FR
7E65AC	CostCenter	Cloud
7E65AC	FK_Manufact	9
237561	CountryEN	NL
237561	CostCenter	Cloud
237561	FK_Manufact	9

3.3 – The Process Workflow (Access Mapping)



3.3 – data processing – finding possible matches

User Access Mapping Table

User	AD-Grp	Rule-Name	Attribute	Value
John Doe	Finance	CountryAndCC	Country	DE
John Doe	Finance	CountryAndCC	Country	FR
John Doe	Finance	CountryAndCC	CostCenter	Software
John Doe	Sales	Country	Country	DE
John Doe	Sales	Country	Country	FR
Mary Doe	Global	CostCenter	CostCenter	Finance
Mary Doe	Global	CostCenter	CostCenter	Finance

(DE =>) GE = GE

Attribute	TableName	ColumnName	CodeMapping
Country	Sales	CountryEN	Country_DE_EN

Sales Fact DimensionMapping Table

DimAccessKey	ColumnName	ColumnValue
43AD56	CountryEN	GE
43AD56	CostCenter	Software
43AD56	FK_Manufact	2
9874FE	CountryEN	GE
9874FE	CostCenter	Software
9874FE	FK_Manufact	7
7E65AC	CountryEN	FR
7E65AC	CostCenter	Cloud
7E65AC	FK_Manufact	9
237561	CountryEN	NL
237561	CostCenter	Cloud
237561	FK_Manufact	9

CodeMapping	MapFrom	MapTo
Country_DE_EN	DE	GE

3.3 – data processing – validating matches

User Access Mapping Table

User	AD-Grp	Rule-Name	Column	Value
John Doe	Finance	CountryAndCC	**CountryEN**	**GE**
John Doe	Finance	CountryAndCC	**CountryEN**	FR
John Doe	Finance	CountryAndCC	CostCenter	Software
John Doe	Sales	Country	**CountryEN**	**GE**
John Doe	Sales	Country	**CountryEN**	FR

User	AD-Grp	Rule-Name	DimAccessKey	Matches
John Doe	Finance	CountryAndCC	43AD56	2 / 2
John Doe	Finance	CountryAndCC	9874FE	2 / 2
~~John Doe~~	~~Finance~~	~~CountryAndCC~~	~~7E65AC~~	1 / 2
~~John Doe~~	~~Finance~~	~~CountryAndCC~~	~~237561~~	0 / 2
John Doe	Sales	Country	43AD56	1 / 1
John Doe	Sales	Country	9874FE	1 / 1
John Doe	Sales	Country	7E65AC	1 / 1
~~John Doe~~	~~Sales~~	~~Country~~	~~237561~~	0 / 1

Sales Fact DimensionMapping Table

DimAccessKey	ColumnName	ColumnValue
43AD56	CountryEN	GE
43AD56	CostCenter	Software
43AD56	FK_Manufact	2
9874FE	CountryEN	GE
9874FE	CostCenter	Software
9874FE	FK_Manufact	7
7E65AC	CountryEN	FR
7E65AC	CostCenter	Cloud
7E65AC	FK_Manufact	9
237561	CountryEN	NL
237561	CostCenter	Cloud
237561	FK_Manufact	9

3.3 – Optimization (1)

- Fast Insert-only preferred updates where possible for intermediate and final mapping tables

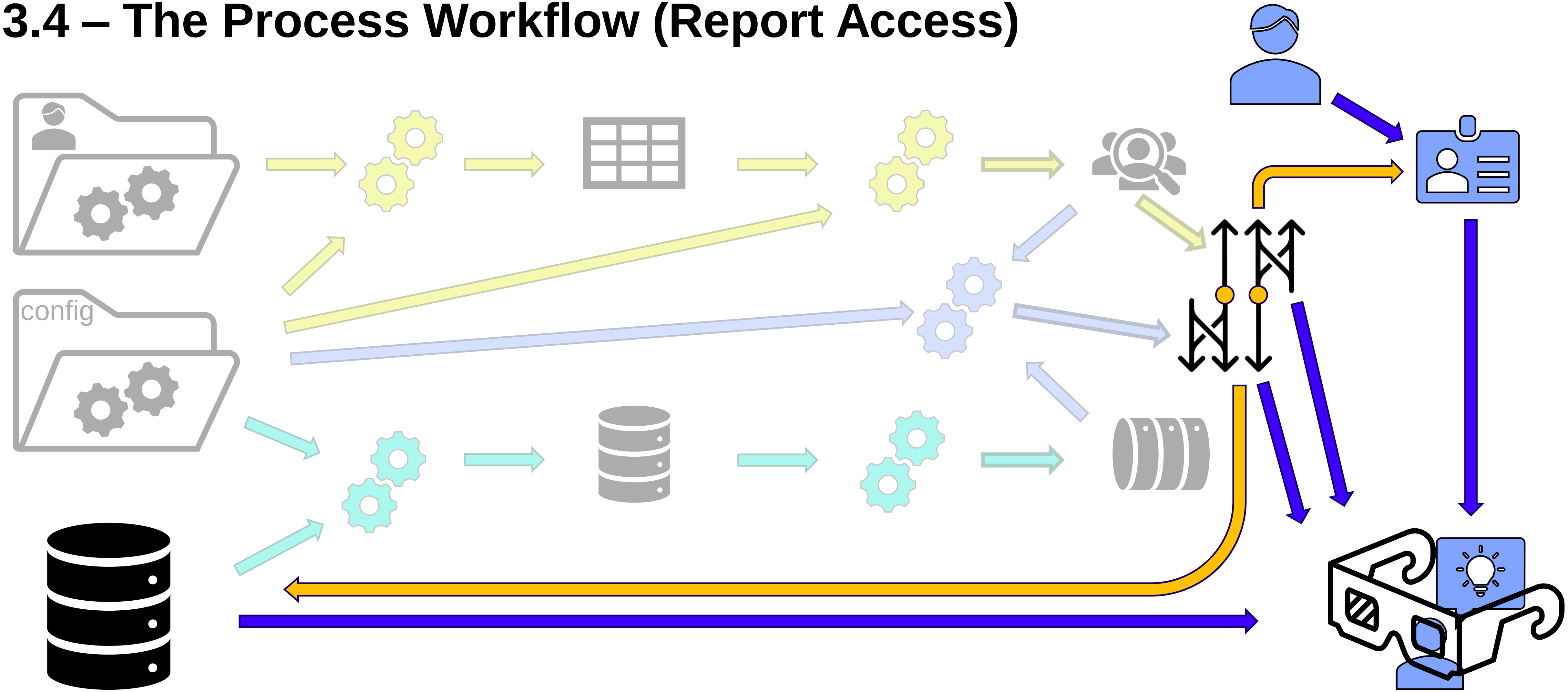
- Users with similar (e.g. global, non-restricted) access should be threated equally.
  - Mapping User ⬄ Fact is split up into
    - User ⬄ SameAccessKey
    - SameAccessKey ⬄ Fact
  - SameAccessKey: Hash-Aggregation segmented by User and AD-Group
    - Rule-Name,
    - Attribute-Name,
    - Attribute-Value

3.3 – Optimization (2)

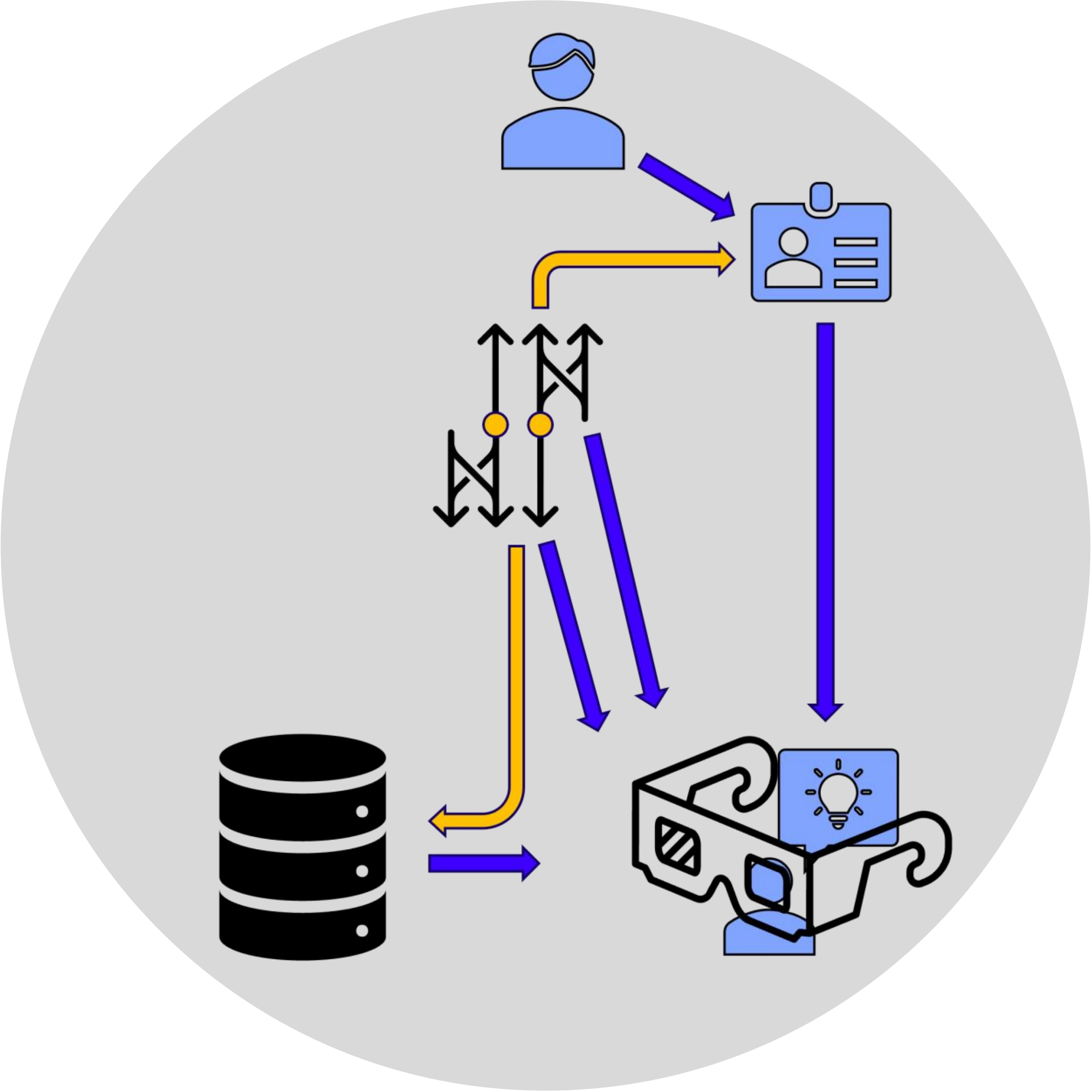
- Users with the same AD-Group and where all the RLS and CLS restrictions are the same
  … get the same SameAccessKey
  - e.g. if the CEO and the CFO have global access to finance data.

- Results:
  - 80% space savings for the mapping
    - Side effect: more frequent cache hits => increased report performance
  - 50% processing time savings
  - 10% overhead during report runtime
    for eliminating duplicate access with multiple AD groups

3.4 – The Process Workflow (Report Access)



Handling complex Row- and Column-Level-Security at scale in Databricks



Conclusion

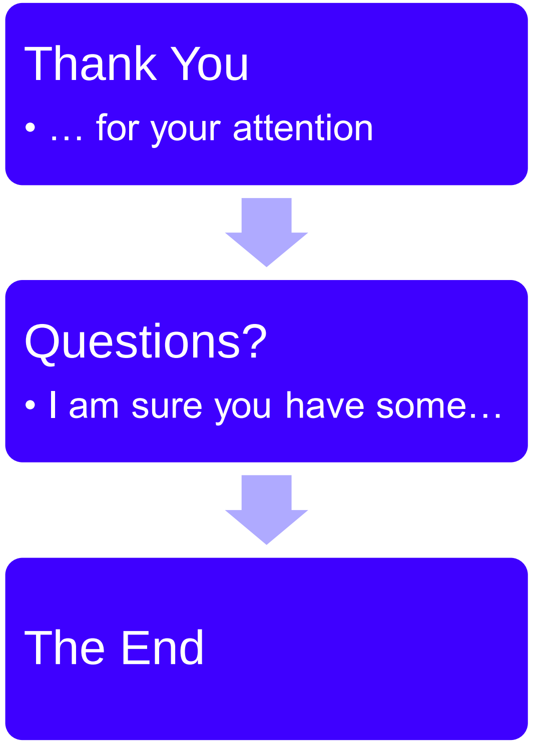
Project goals reached:

- Efficient pre-processing of RCLS rules
- Efficient view logic for report access

Next Steps:

- Aggregation Levels for CLS
- Spread the word

Handling complex Row- and Column-Level-Security at scale in Databricks



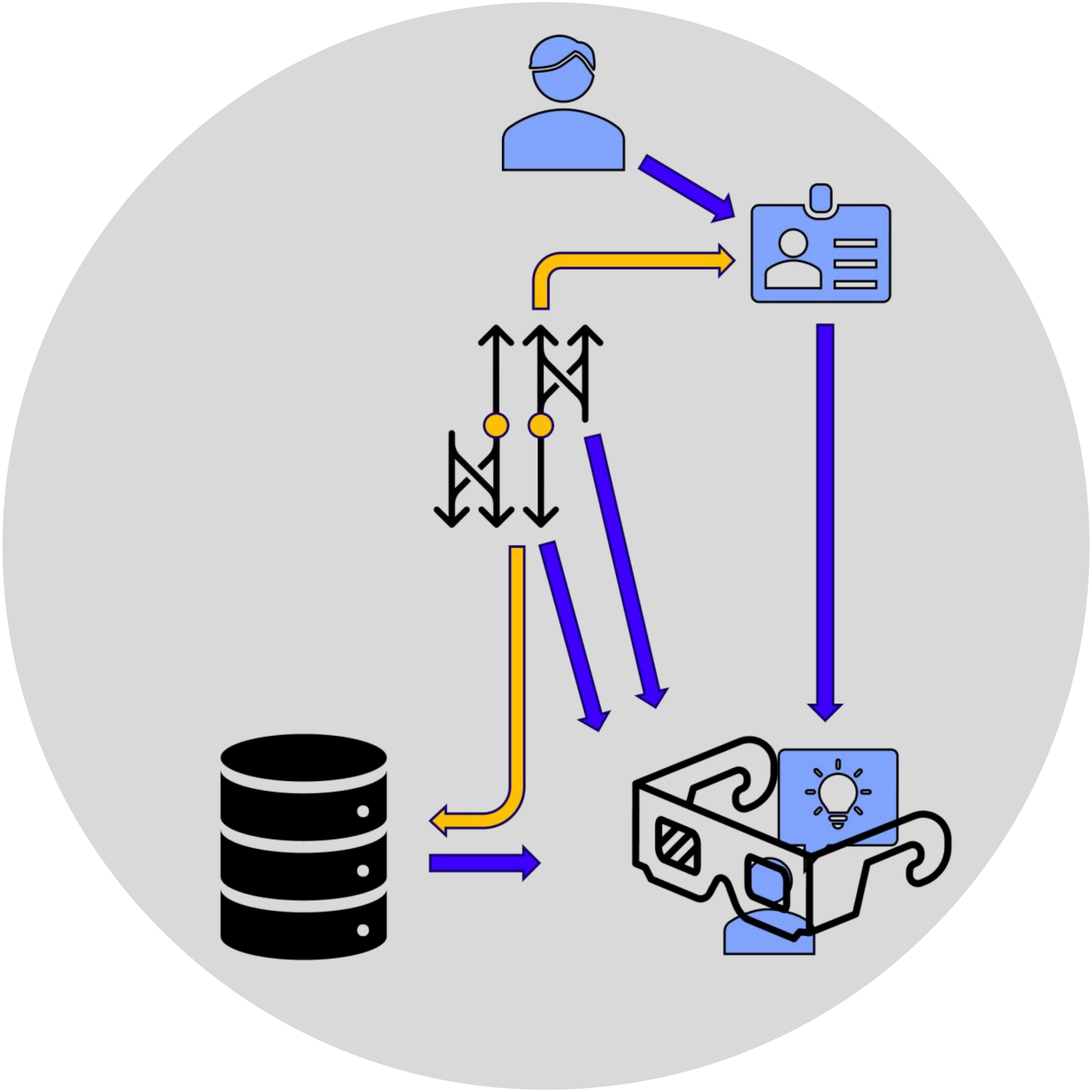
Thank You

- … for your attention

Questions?

- I am sure you have some…

The End



Disclaimer

This publication contains proprietary information that is protected by copyright. SoftwareOne reserves all rights thereto.

SoftwareOne shall not be liable for possible errors in this document. Liability for damages directly and indirectly associated with the supply or use of this document is excluded as far as legally permissible.

The information presented herein is intended exclusively as a guide offered by SoftwareOne. The publisher's product use rights, agreement terms and conditions and other definitions prevail over the information provided herein. The content must not be copied, reproduced, passed to third parties or used for any other purposes without written permission of SoftwareOne