SVENSKA SPEL

Journey in country of data access governance

Magnus.Runesson@svenskaspel.se

Hortonworks Stockholm Summit 2018-12-06

Who is talking?

Magnus
Runesson

Developer
Ops
RDBMS
BigData
High performance

Data Engineer @ Svenska Spel

SVENSKA SPEL

Gaming is for everyone´s enjoyment

Agenda

Why?

Svenska Spel’s data warehouse

Atlas & Ranger

How did we implement it?

Learnings

Conclusions

Why?

GDPR requires

- clear purpose for PII data
- privacy by design
- clear consent or legal ground
- not to use/store PII if not needed
- people own their own data.
- penalty if not followed

New gaming market requires

- introduce multi tenancy

Goals

Our customers and partners integrity is protected

Follow competition regulation

Users have only access to data aimed for current purpose

Keep doing our required processing

Adaptable for new requirements

Maintainable solution

Svenska Spel’s data warehouse

Svenska Spel’s data warehouse

Moved from classic Cognos + Oracle

HDP 2.6 using Hive

Includes Personal Identifiable Information (PII)

300+ event streams in

150+ published tables and views

Model based development

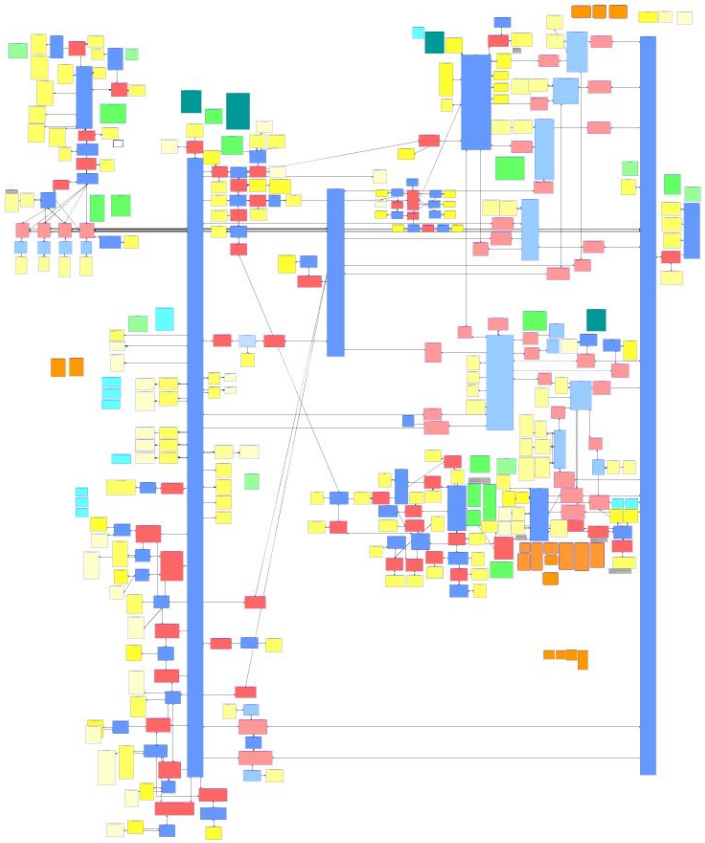
Used data are

- Understood
- Documented
- Modelled

Modelled with Data Vault

Oracle SQL Developer Data Modeler

SQL code generated from model



Data Vault

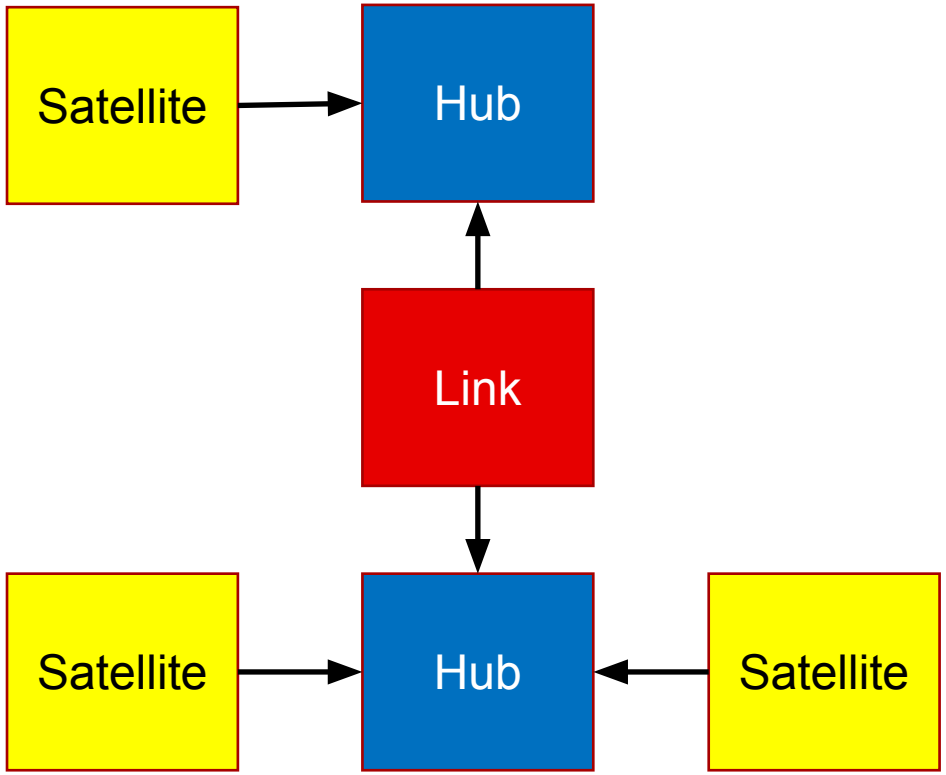
History tracking

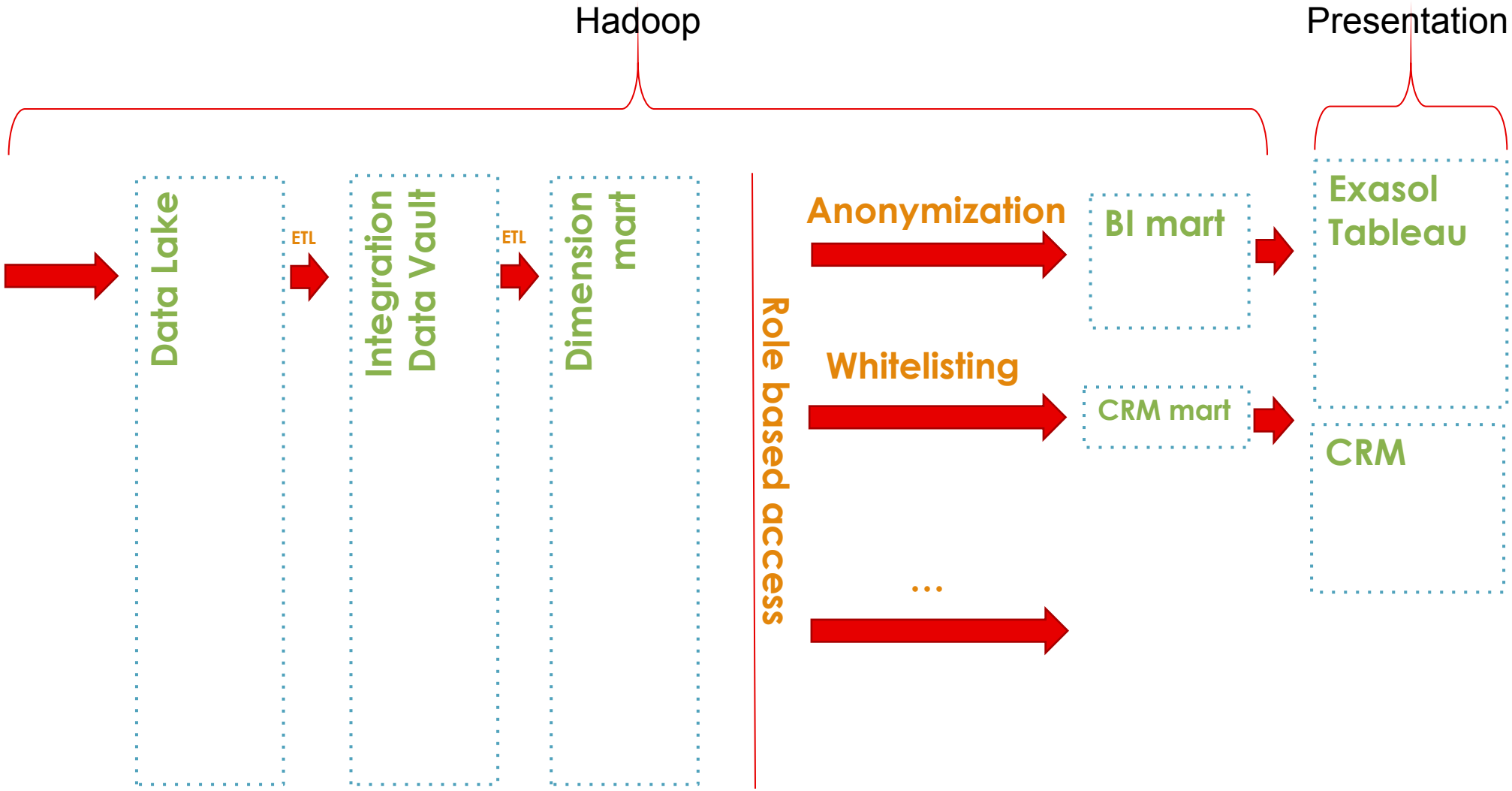
Uniquely linked

Pattern based

Easy to generate code

Easy to add new sources




Hadoop
Presentation
Data Lake
ETL
Integration Data Vault
ETL
Dimension mart
Role based access
Anonymization
BI mart
Whitelisting
CRM mart
...
Exasol Tableau
CRM

Apache Atlas and Ranger

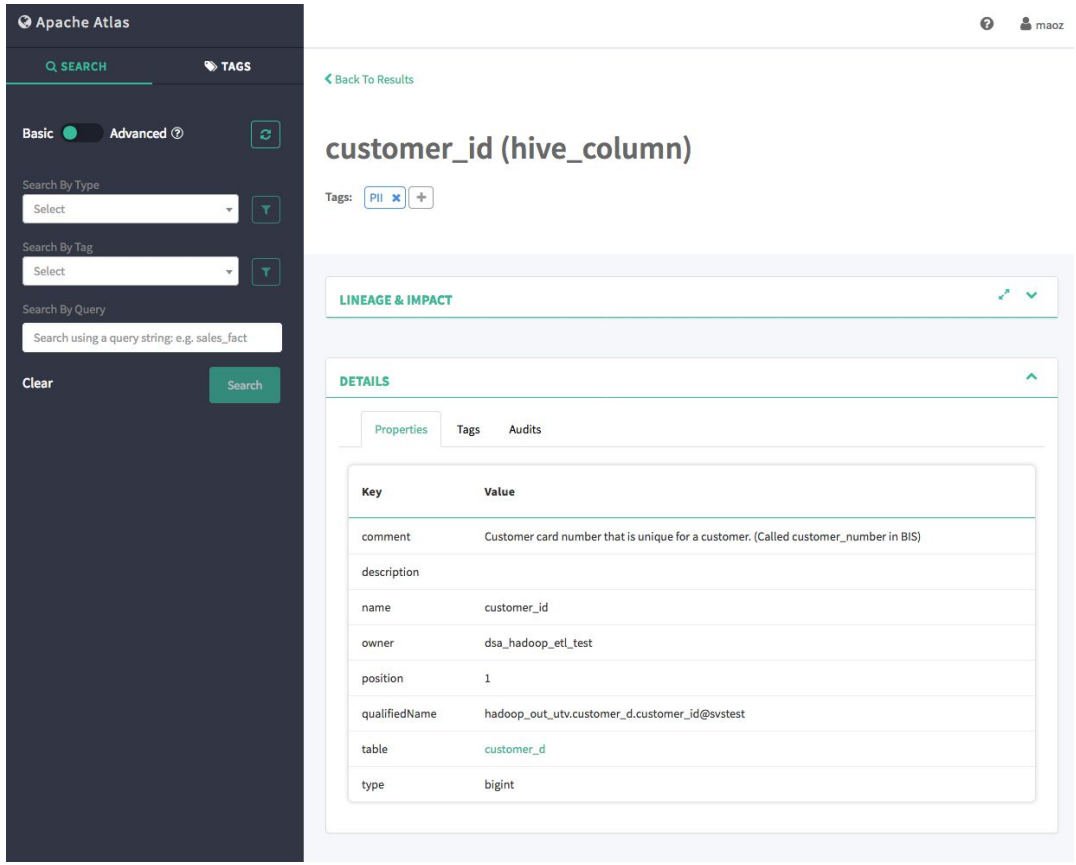
Apache Atlas

Metadata about resources

Resource is

- Table
- Column
- Schema
- File on HDFS
- …

Lineage



Atlas tags

Tags have no meaning themselves

Your business vocabulary define the meaning

Example of tags:

- Business entity owning the data
- Indication of sensitive data

The rules in Ranger enforces the policy

Separate metadata from policy implementation

Apache Ranger

Is user U allowed to do operation O on resource R?

Access

Row based filtering

Masking

Audit logging

Resources referred with tags

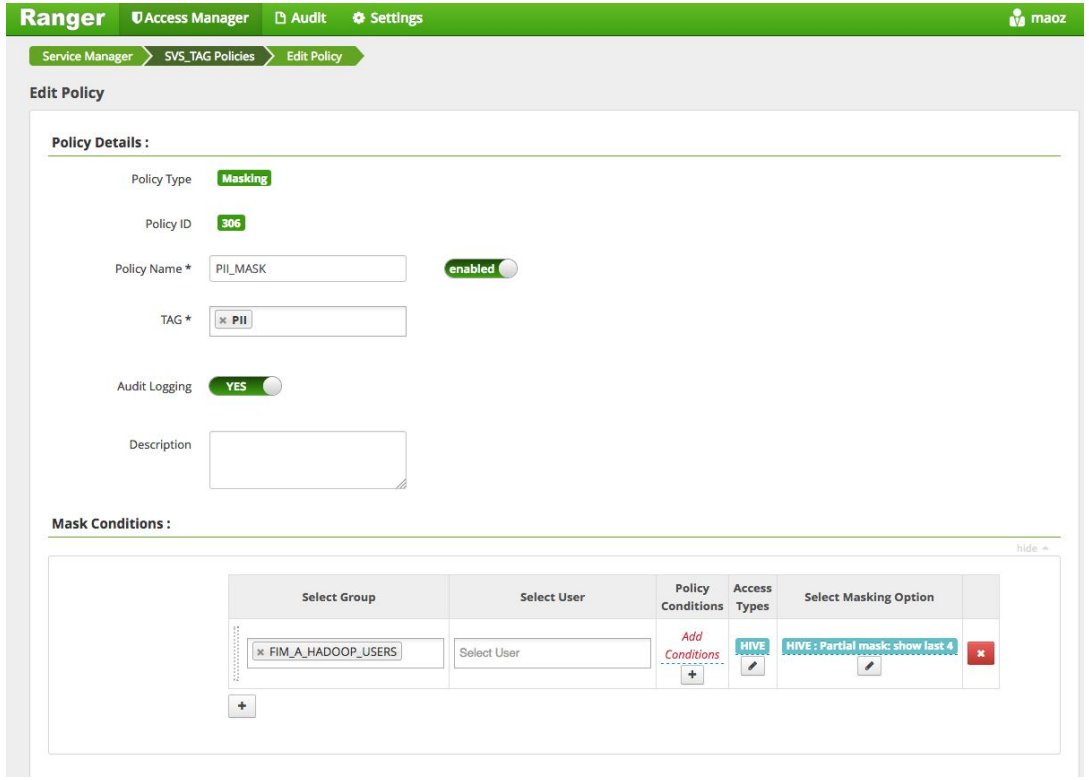


Table in Hive before we started our work

customer

Customer_id	Name	Postal_code	Has_phone	Marketing
1	Steve	12345	False	False
2	Bill	54321	True	False
3	Paul	54672	False	True

Add PII tags on table and columns in Atlas.
No behaviour change.

customer

PII_table

Customer_id PII	Name PII	Postal_code	Has_phone	Marketing
1	Steve	12345	False	False
2	Bill	54321	True	False
3	Paul	54672	False	True

We set a rule in Ranger to mask PII columns
Analyst view

customer

PII_table

Customer_id PII	Name PII	Postal_code	Has_phone	Marketing
17	ABC	12345	False	False
42	DEF	54321	True	False
13	BDE	54672	False	True

Ranger restrict our CRM user to only see rows with Marketing = True

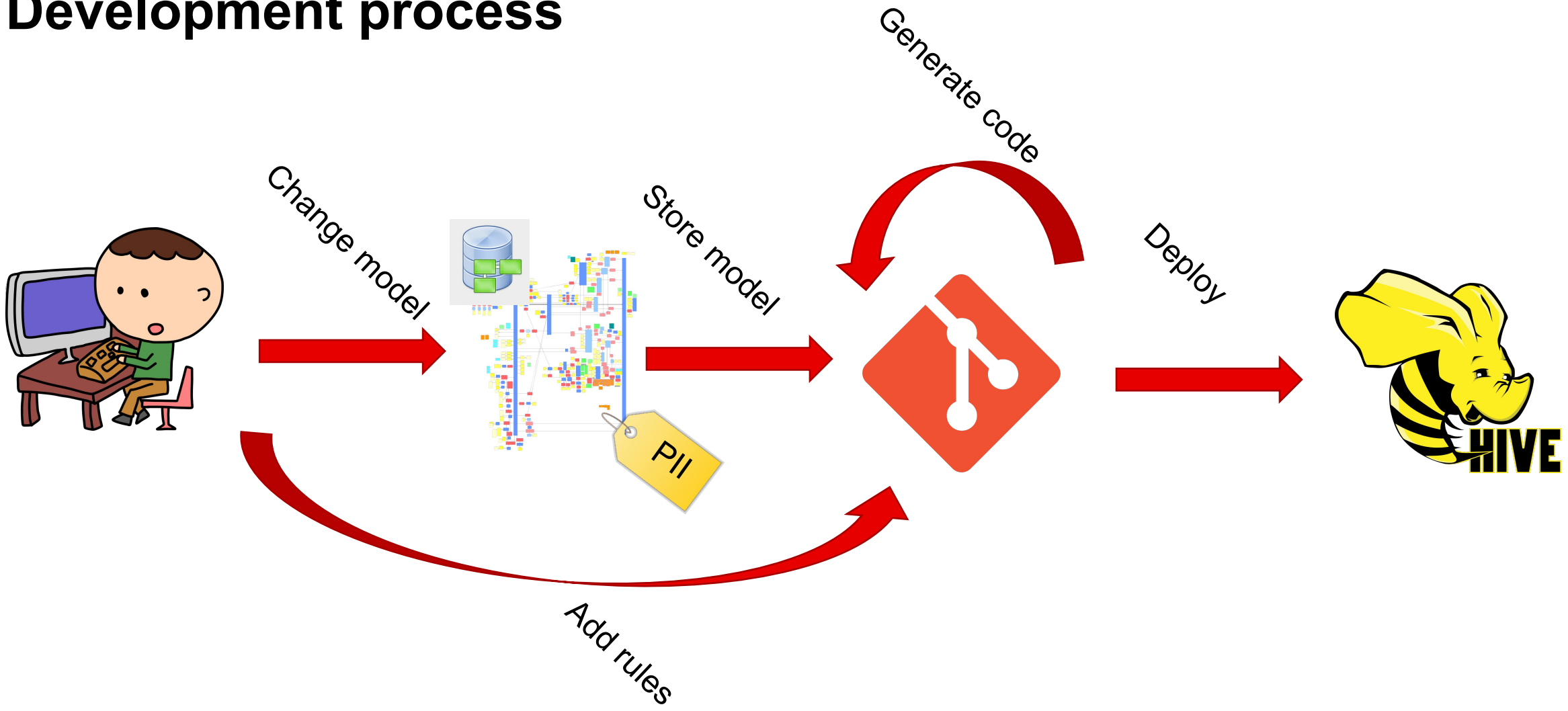
customer

PII_table

Customer_id PII	Name PII	Postal_code	Has_phone	Marketing
3	Paul	54672	False	True

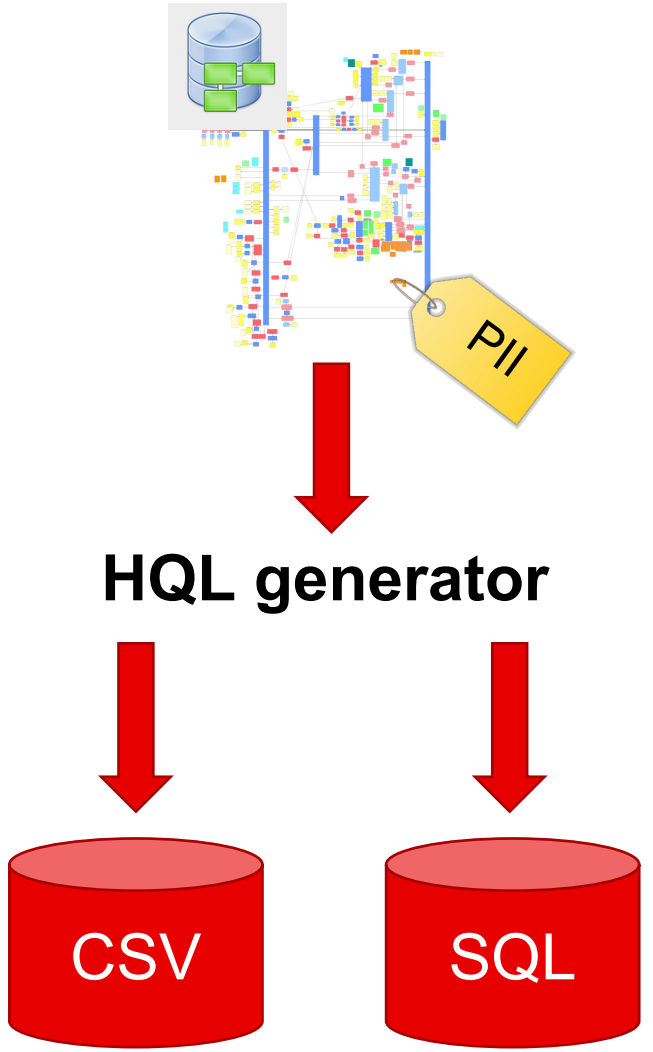
How did we implement this?

Development process



HQL generator

- In-house tool
- Template based generation of SQL/HQL
- Generate files with tag-information
  - Tables and columns respectively



Tag file for columns

```
schema;table;attribute;tags

dim_mart;customer_d;customer_id;PII,Sensitive

dim_mart;customer_d;has_phone;
```

Corresponding file for tables without attribute(column)

Ranger rules

Hand coded of rules per tag

Policy tool applies rule on all tables with the tag

Can be different rules for different users

Filter gets appended to where condition by Ranger

Used for

- Row based filtering (access)
- Masking (anonymization)

Catch all rule to deny access to tables not in our model

Ranger rule filter example

```
{
  "command": "apply_tag_row_rule",
  "filters": [
    {
      "groups": [ "tenant_1"],
      "users": [],
      "tagFilterExprs": [
          {
            "tags": [ "multitenant" ],
            "filterExpr": "${table}.tenant_id = 1"
          }
        ]
      },
...
```

Deployment process

*.sql
table_tags.csv
column_tags.csv
ranger_policies.json

Apply *.sql DDL

Policy tool - tag files

Policy tool - policy file

HIVE

Apache Atlas

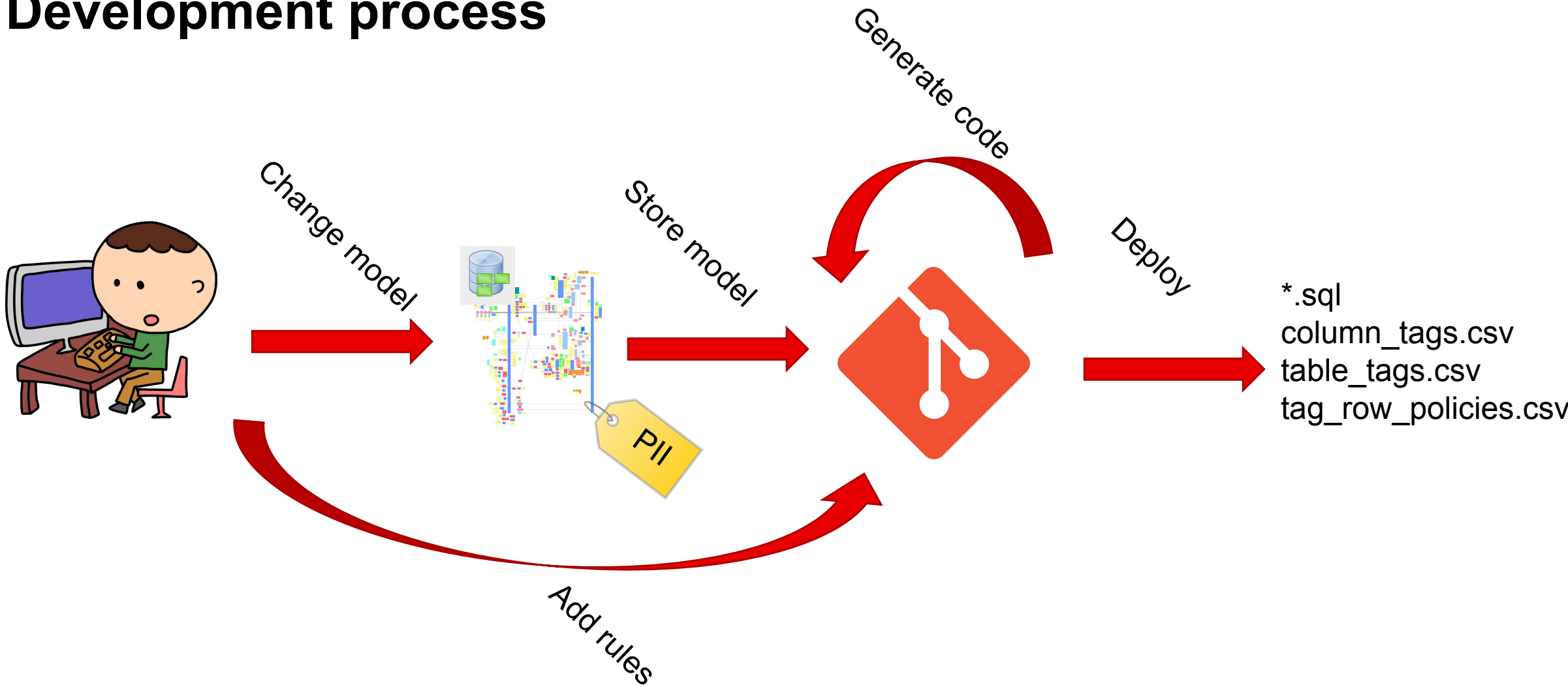
Apache Ranger

Policytool

- Makes it easy to manage
  - Atlas tags
  - Ranger policy rules
- Command line tool
- Consumes tags from CSV files
- Consumes policies from JSON files
- Calls Atlas and Ranger API
- Ensure same access on Hive as HDFS (not filtering and masking)
- Supports tag-based filtering

Put everything together

Development process



Deployment process

*.sql
column_tags.csv
table_tags.csv
ranger_policies.json

Apply *.sql DDL

Policy tool - tag files

Policy tool - policy file

HIVE

Apache Atlas

Apache Ranger

Change in view of an Analyst

Before

customer

Customer_id	Name	Postal_code	Has_phone	Marketing
1	Steve	12345	False	False
2	Bill	54321	True	False
3	Paul	54672	False	True

Analyst

customer PII_table

Customer_id PII	Name PII	Postal_code	Has_phone	Marketing
17	ABC	12345	False	False
42	DEF	54321	True	False
13	BDE	54672	False	True

CRM

customer PII_table

Customer_id PII	Name PII	Postal_code	Has_phone	Marketing
3	Paul	54672	False	True

Learnings

Clear business rules

Work closely with the business

Avoid too complex rules

Minimize number of rules

Use {user}, public and other alias Ranger uses.

Systematic model

People do unconsciously things differently

Keep hdfs and hive rules in sync

Use tags as much as possible

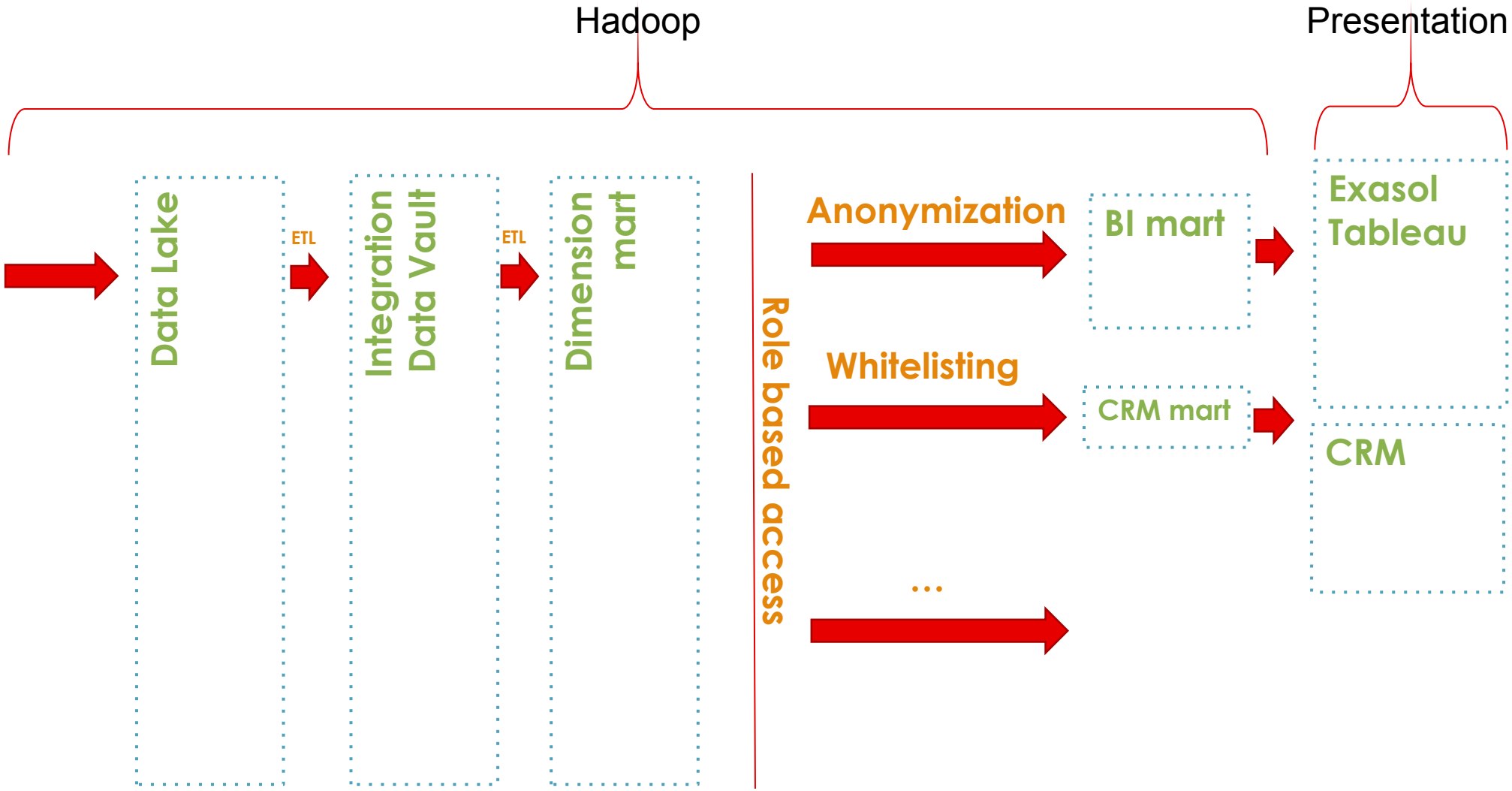
Automate

Ensure rules are in sync with what is deployed

Use CI/CD

Ask HW for latest patches on 2.6.5 (ATLAS-2634, HIVE-20633, ATLAS-2891, ATLAS-2975)

Hey,
would it not be nice to have the same rules in the presentation layer?


Hadoop
Presentation
Data Lake
ETL
Integration Data Vault
ETL
Dimension mart
Role based access
Anonymization
Whitelisting
...
BI mart
CRM mart
Exasol Tableau
CRM

Atlas & Ranger on Exasol

Transfer rules and tags to Exasol

Use virtual schemas to apply them

Reduce amount of data in Exasol

Lower license cost

Single source of truth of access policies

Experiences of Atlas and Ranger

- Simple and easy model
- Limited performance penalty
- Tag on table with masking rule => all columns masked
- Lot of moving pieces
- Hard to understand API doc
- Restriction on Ranger row based filtering (not on tags)
- Row based filtering and masking not on direct file access

Reached Goals

- Our customers and partners integrity is protected
- Users have only access to data aimed for current purpose
- Keep doing our required processing
- Adaptable for new requirements
- Maintainable solution

Conclusions

- Goals reached
- No SQL changes
- Scale when new datasets added
- Our data model is guaranteed in sync
- Better comments in Hive
- Minimal impact on ETL developers workflow

Takeaways

- Make it as simple as possible
- Automate
- Know your tool
- Be clear on your authorization model
- Know your data

Resources

cobra-policytool on GitHub https://github.com/SvenskaSpel/cobra-policytool

SVENSKA SPEL

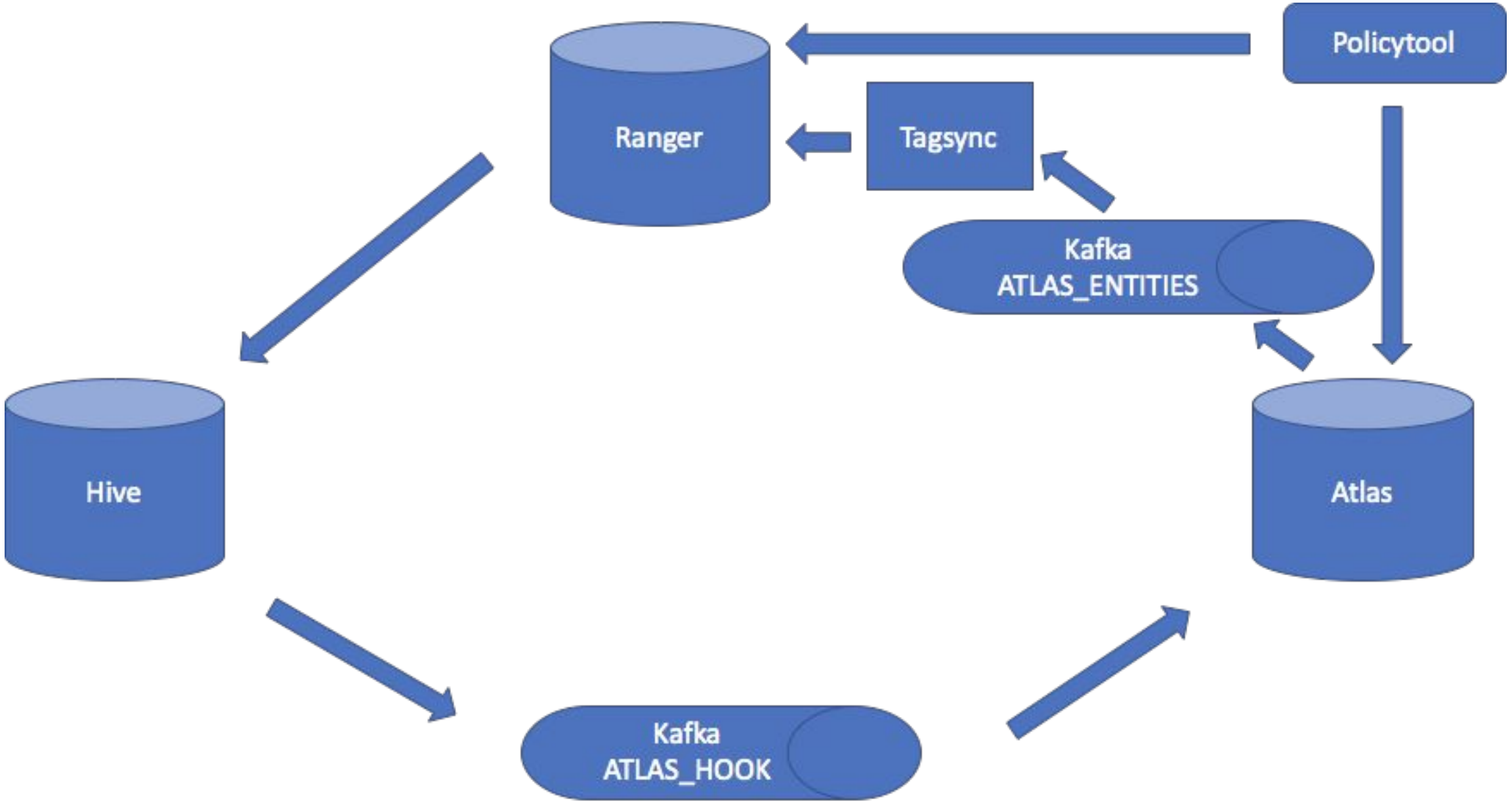
Thank you!

Magnus.Runesson@svenskaspel.se

@MRunesson

karriar.svenskaspel.se

BONUS - How everything is connected


Policytool
Ranger
Tagsync
Kafka
ATLAS_ENTITIES
Hive
Atlas
Kafka
ATLAS_HOOK