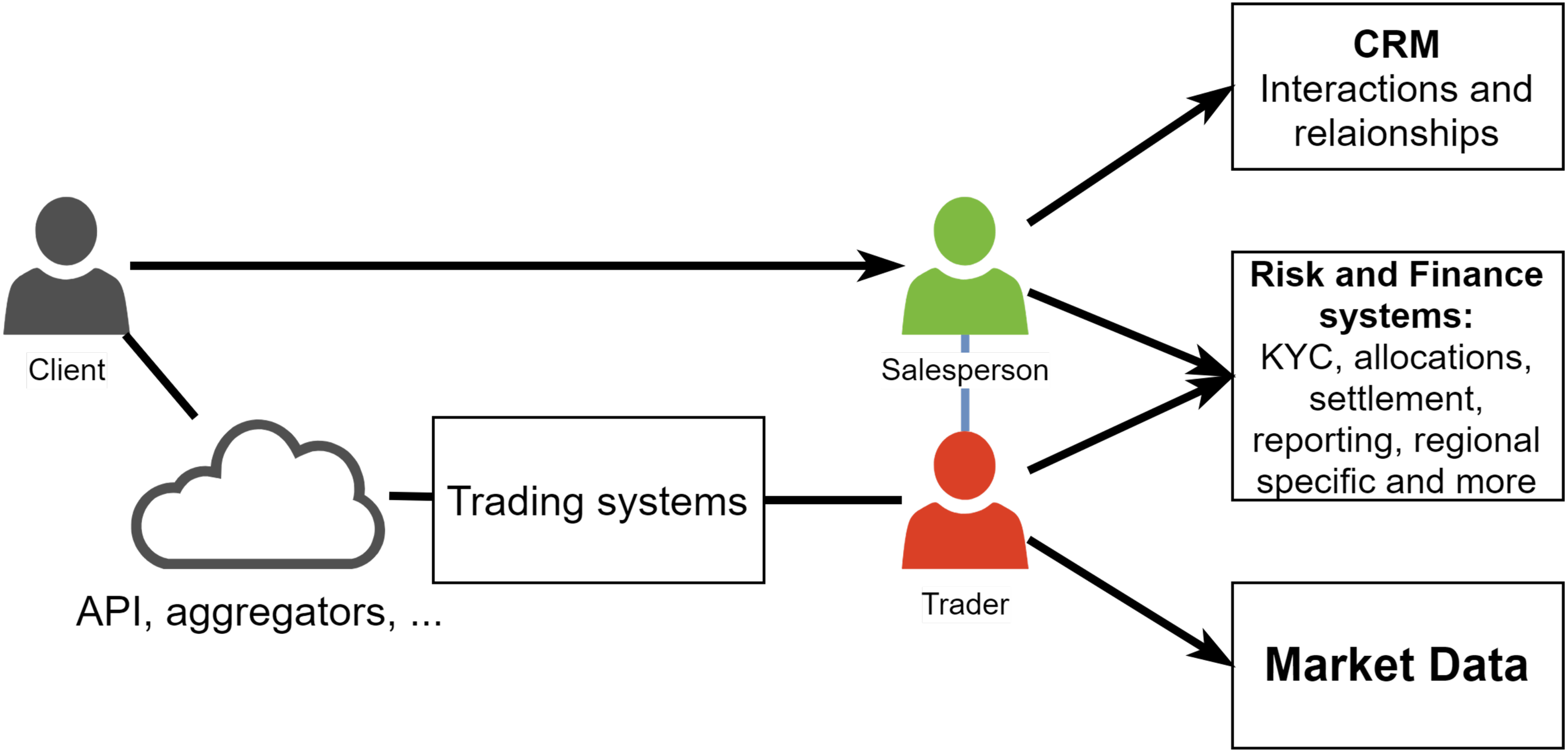
Deutsche Bank
Technology
Centre

Secure interactive big data: Business intelligence on Clickhouse

Outline

- Data in Investment Bank business
- Data warehouse
  - Why?
  - Why ClickHouse?
  - Data driven access control: ABAC & SQL Hardening
- Costly errors to be avoided
- A couple of epic wins

Data in Investment Bank



Client

API, aggregators, ...

Trading systems

Salesperson

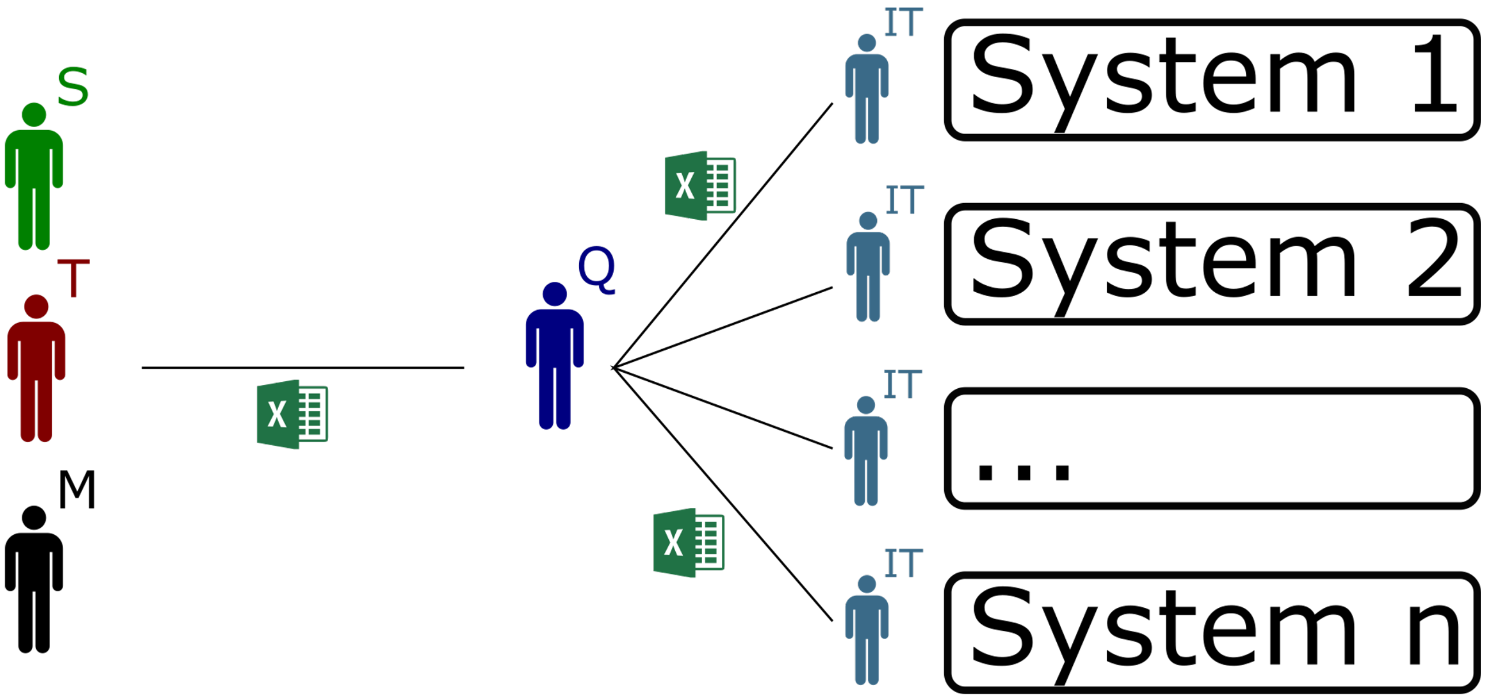
Trader

CRM
Interactions and relaionships

Risk and Finance systems:
KYC, allocations, settlement, reporting, regional specific and more

Market Data

The Need



S - Sales People
T - Traders
M - Management

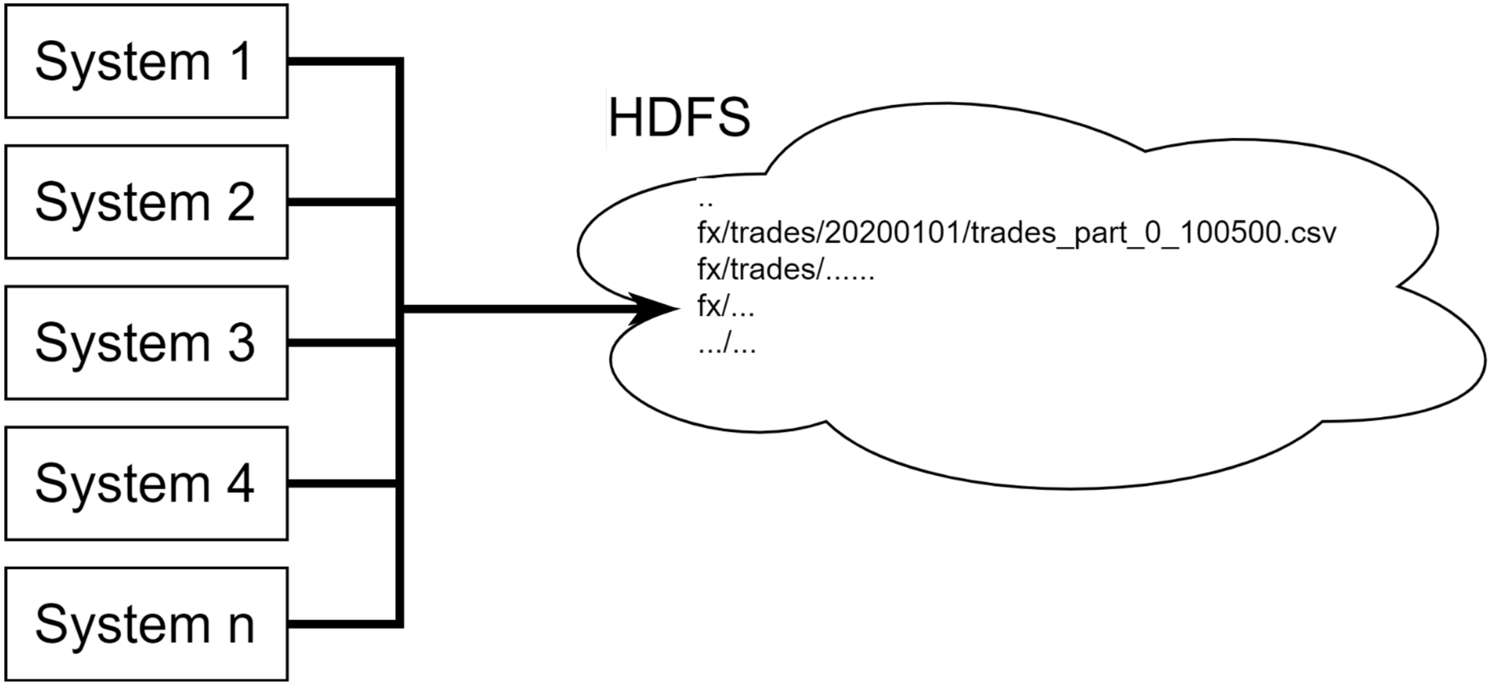
Requirements and Constraints

- All data in one place
  - Compliant
  - With free-hand SQL written by analysts
  - Over Big Data
- Fast time to market
  - Fast prototyping
  - Path to prod for successful reports

KDB? Hadoop? Vertica? Teradata? Greenplum?

Nope. Clickhouse!

Hadoop / Classic data lake



System 1

System 2

System 3

System 4

System n

HDFS

..
fx/trades/20200101/trades_part_0_100500.csv
fx/trades/......
fx/...
.../...

- Data quality is poor
- Hard for analysts
- SME’s/owners are forced to work with hadoop

KDB

1. Steep learning curve
2. Single-threaded, requires infrastructure.
3. Easy to overload

Problem statement: count number of open trades in a specified book.

KDB query:

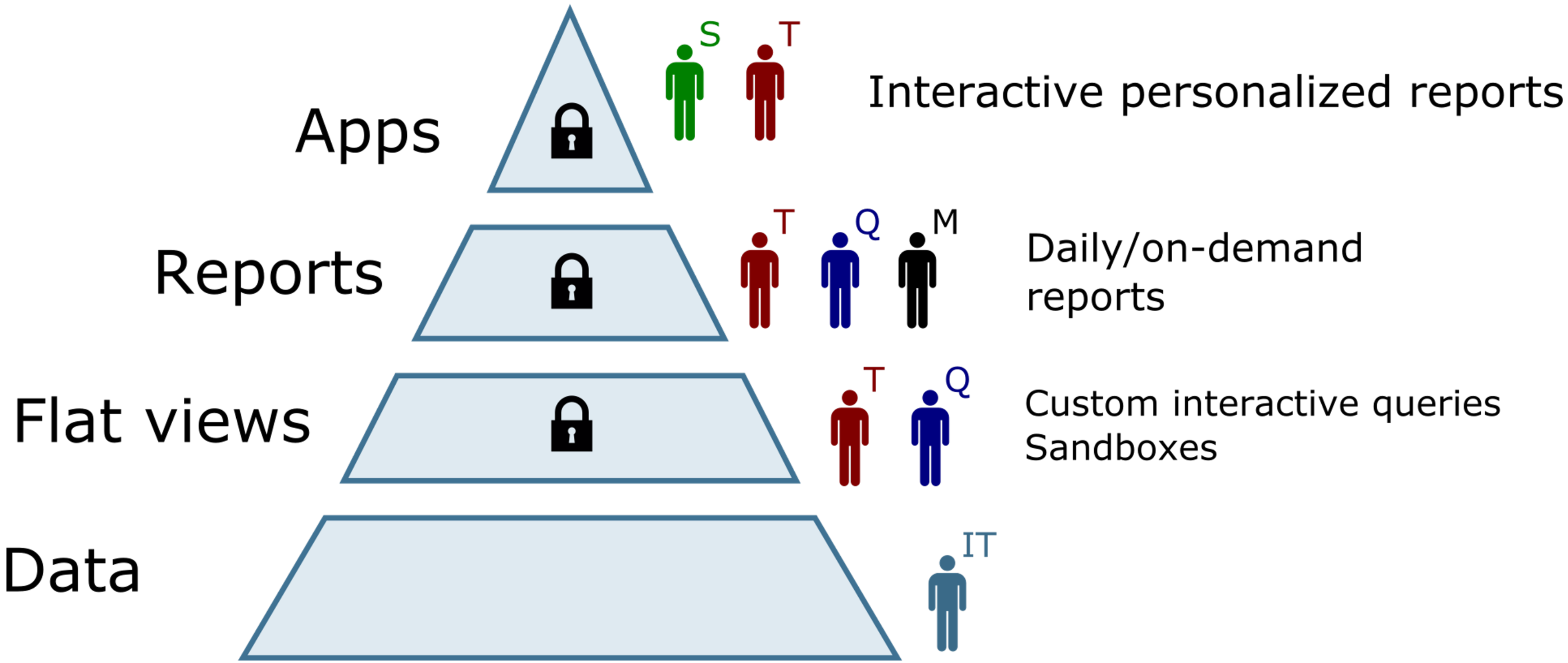
```
select count i from trades where date=2016.12.01, version =
(max;version) fby systemid, status=`Open, book=`LNFEUUS

165

select count i from trades where date=2016.12.01, status=`Open,
book=`LNFEUUS, version = (max;version) fby systemid

327
```

Clickhouse is on par with KDB

Table size 75 Mln rows

Case	ClickHouse	KDB 1 thread	KDB 16 slaves
Simple average	64	370	300
Vector calculations	143	2222	677
Simple group by	136	451	297
Group by partition / index	41	74	108
Group by with where	144	740	548
fby	1109	14765	2185

The Users



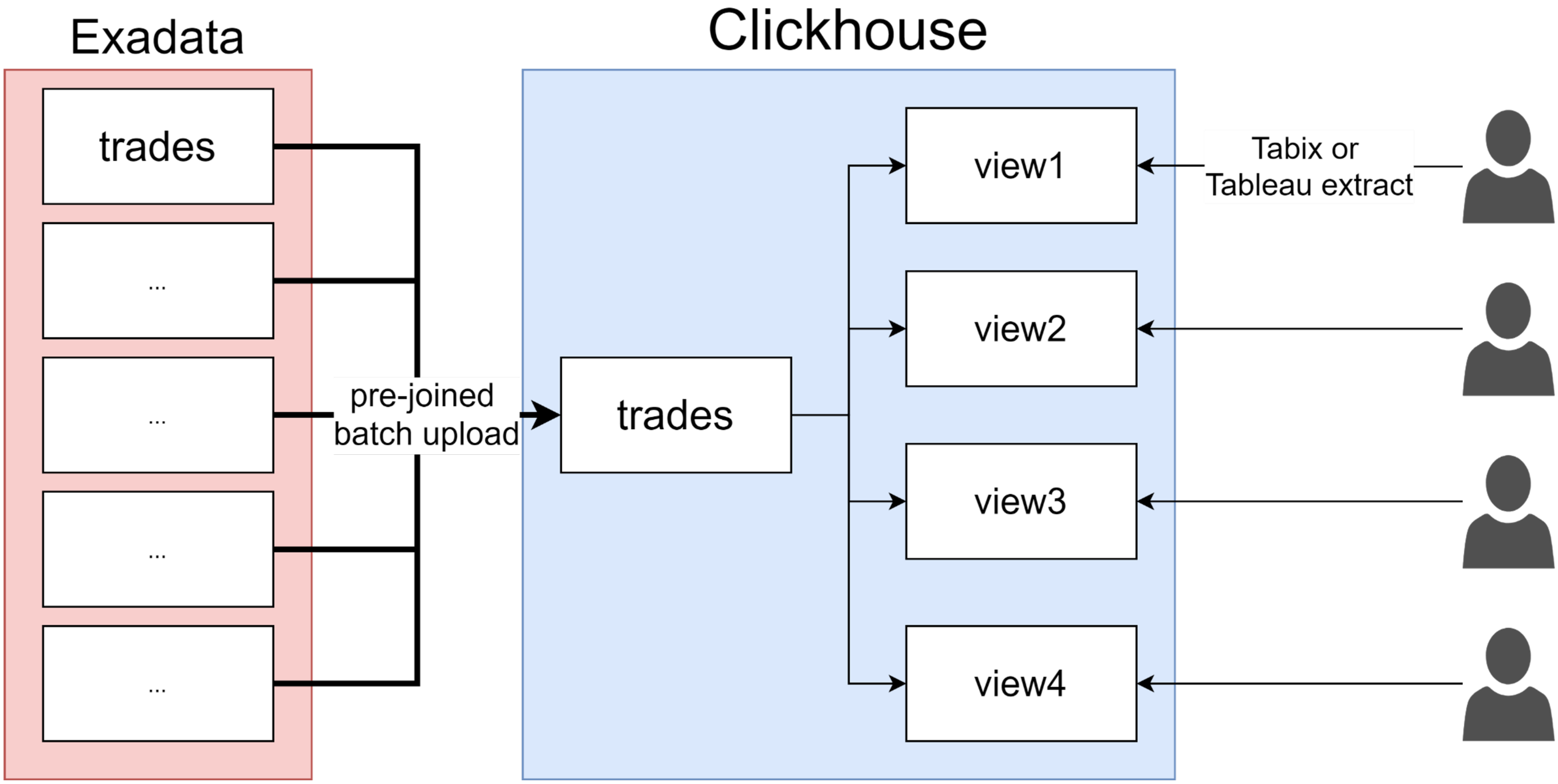
Apps – Interactive personalized reports

Reports – Daily/on-demand reports

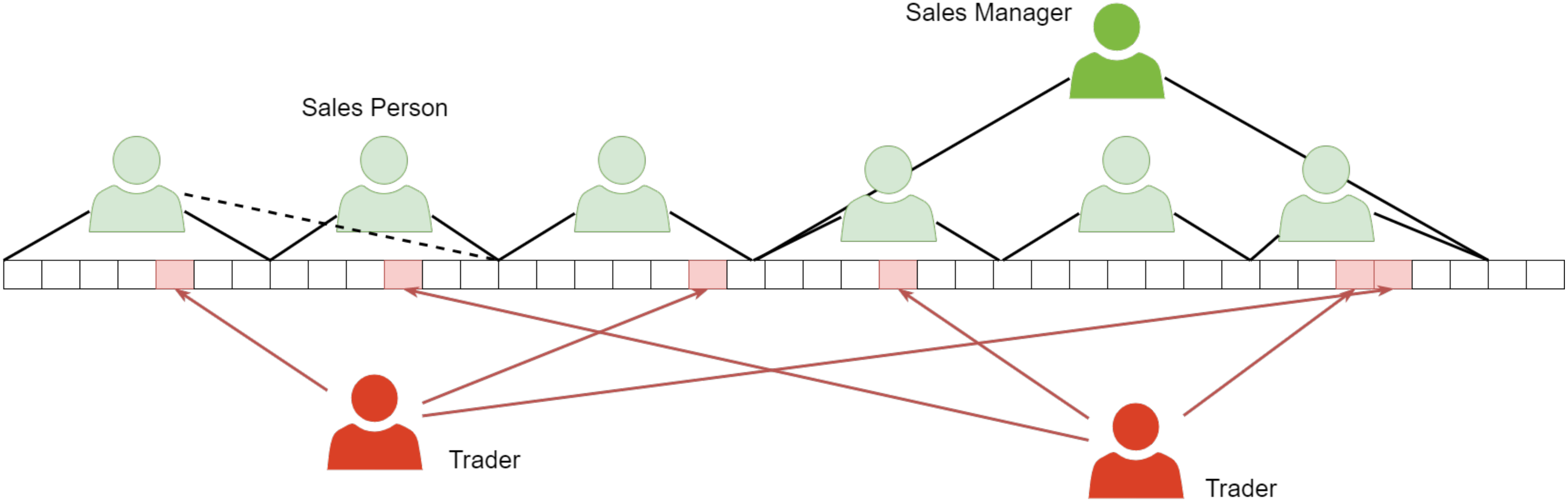
Flat views – Custom interactive queries
Sandboxes

Data

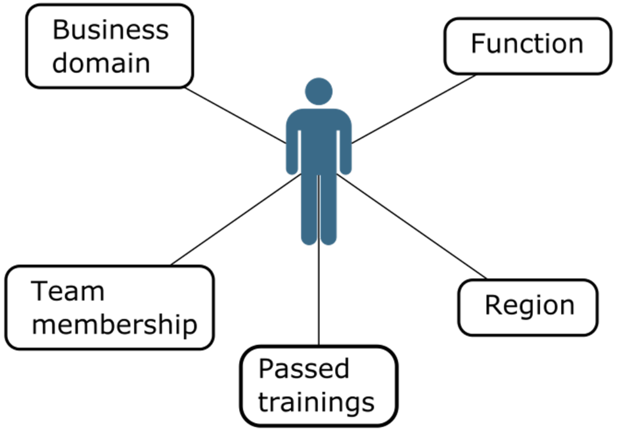
DWH on ClickHouse – first version

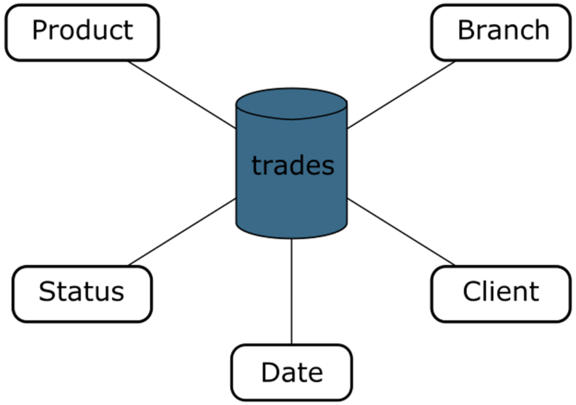


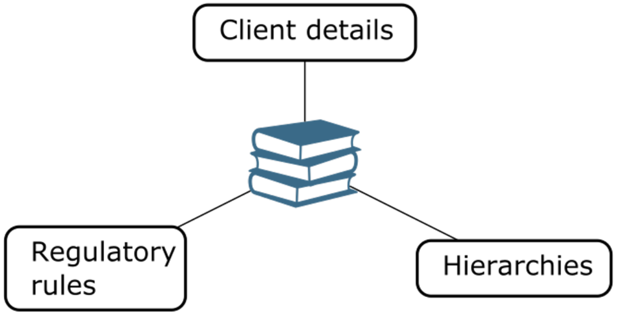
Access Control



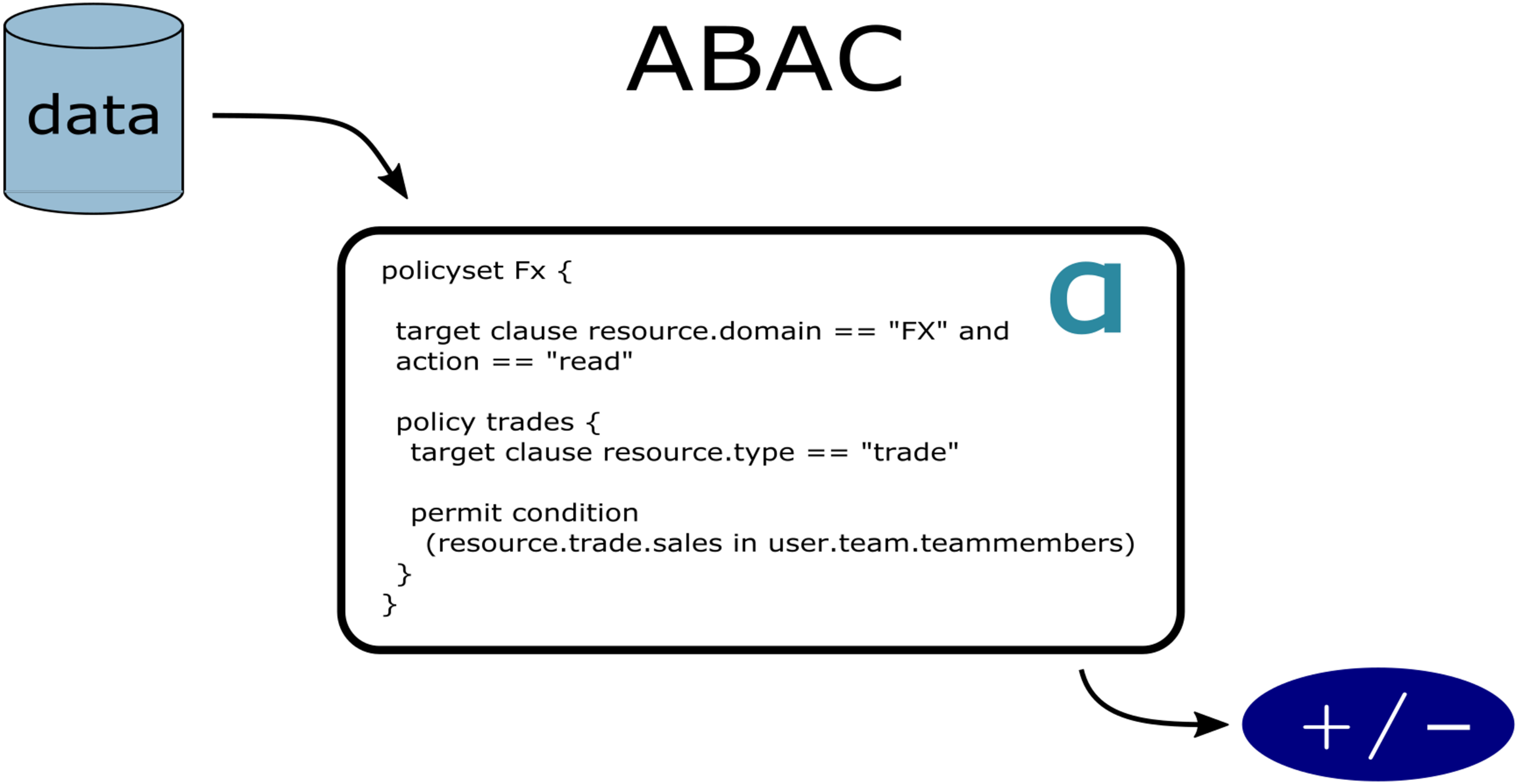
Attribute based access control





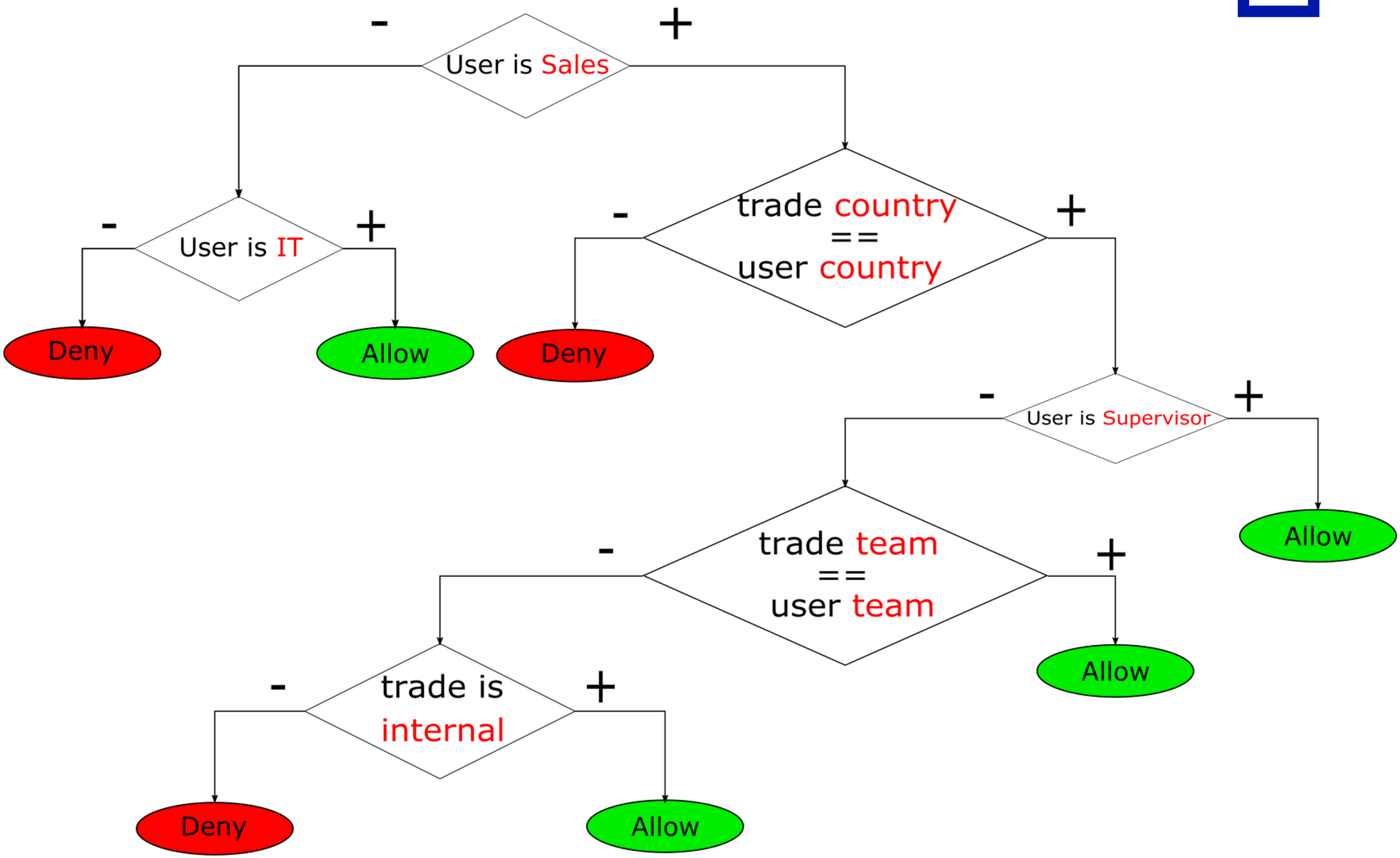


ABAC

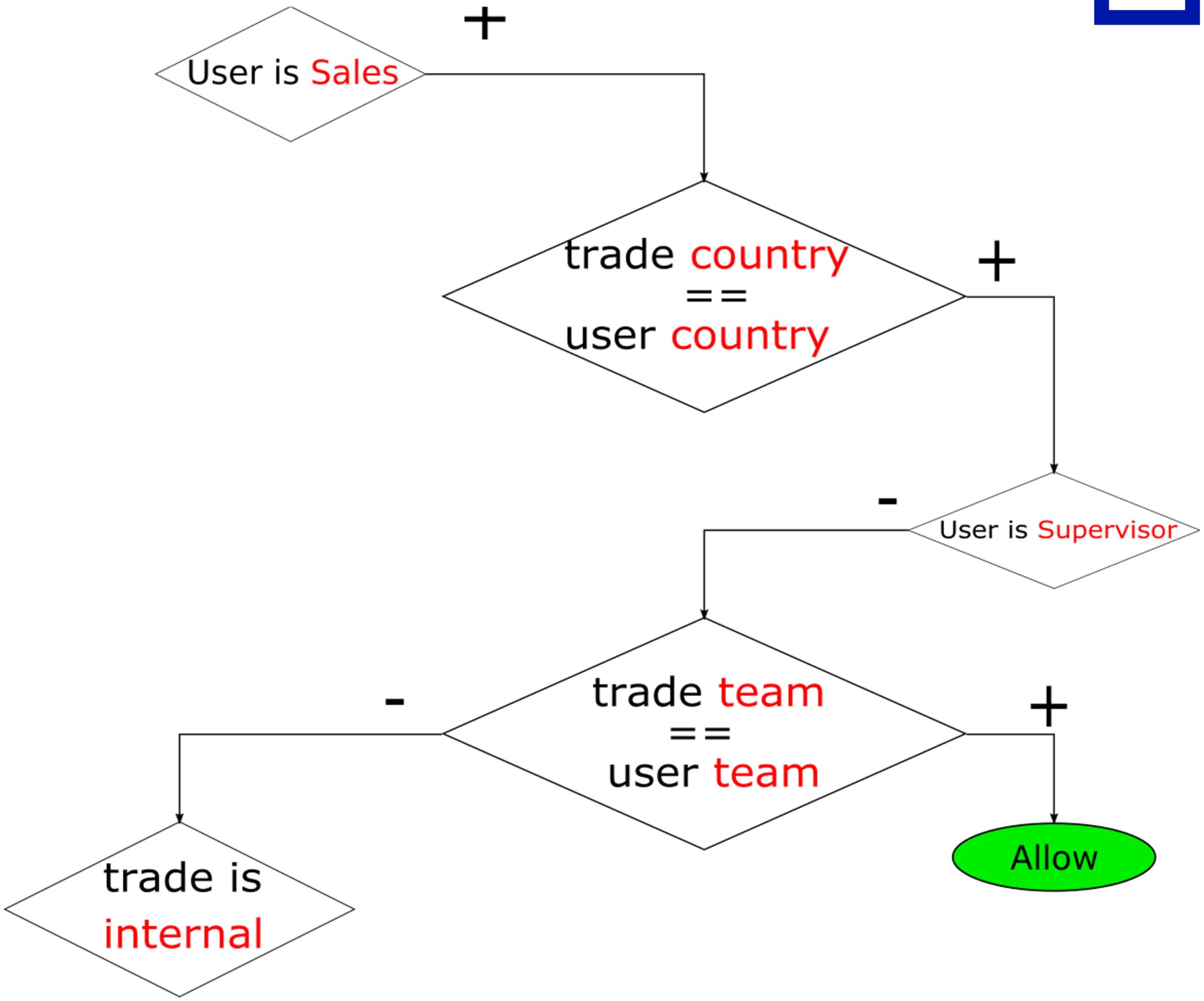


```
policyset Fx {

 target clause resource.domain == "FX" and
 action == "read"

 policy trades {
  target clause resource.type == "trade"

  permit condition
   (resource.trade.sales in user.team.teammembers)
 }
}
```

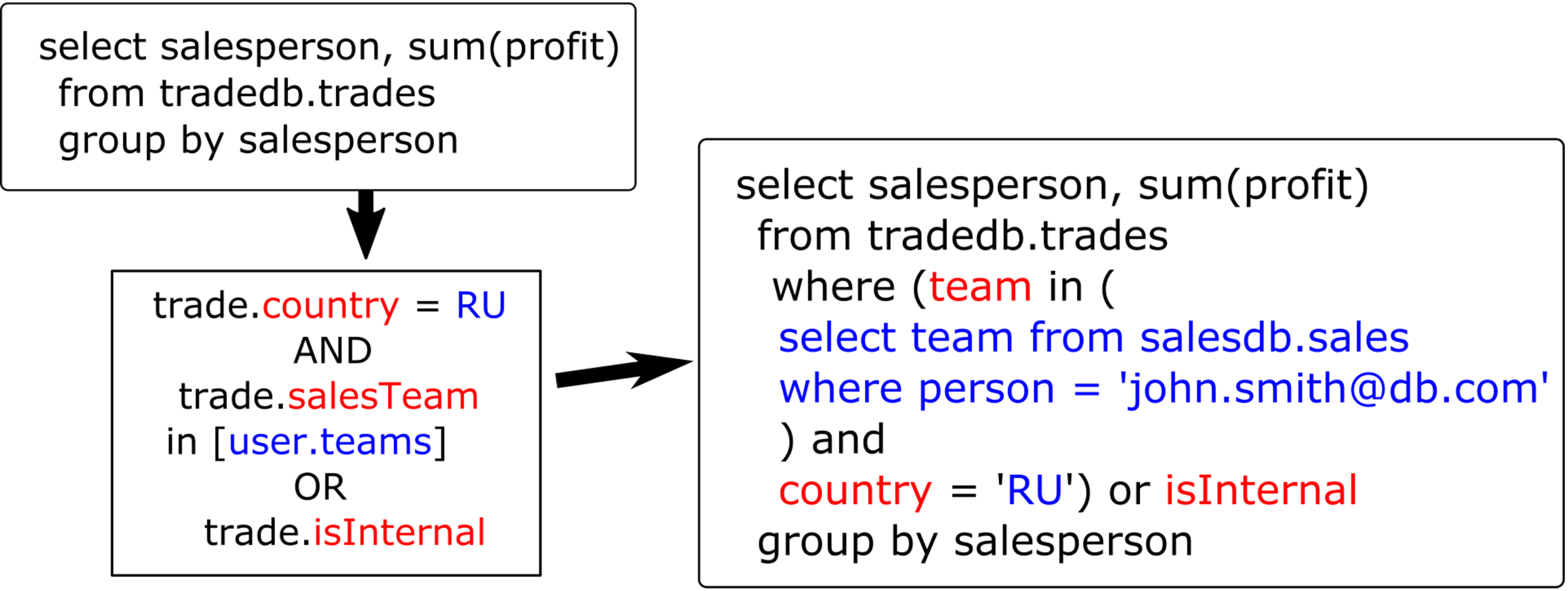
https://en.wikipedia.org/wiki/ALFA_(XACML)


-
+
User is Sales
-
+
User is IT
Deny
Allow
-
trade country
==
user country
+
Deny
-
User is Supervisor
+
Allow
-
trade team
==
user team
+
Allow
-
trade is
internal
+
Deny
Allow

trade.country = RU
AND
trade.salesTeam
in [team1, team2]
OR
trade.isInternal



SQL AST transformation



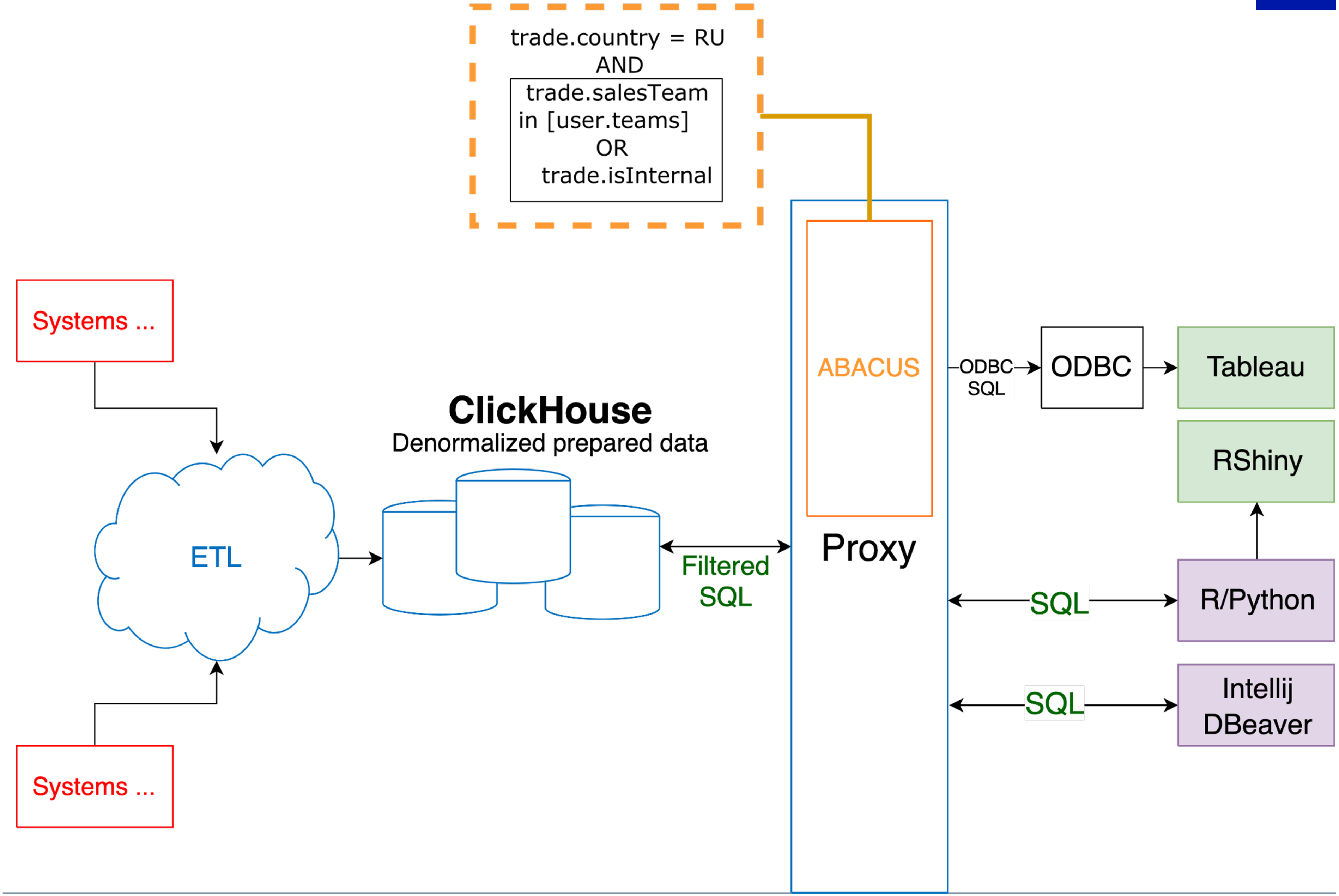
```
select salesperson, sum(profit)
 from tradedb.trades
 group by salesperson
```

```
trade.country = RU
      AND
 trade.salesTeam
in [user.teams]
      OR
  trade.isInternal
```

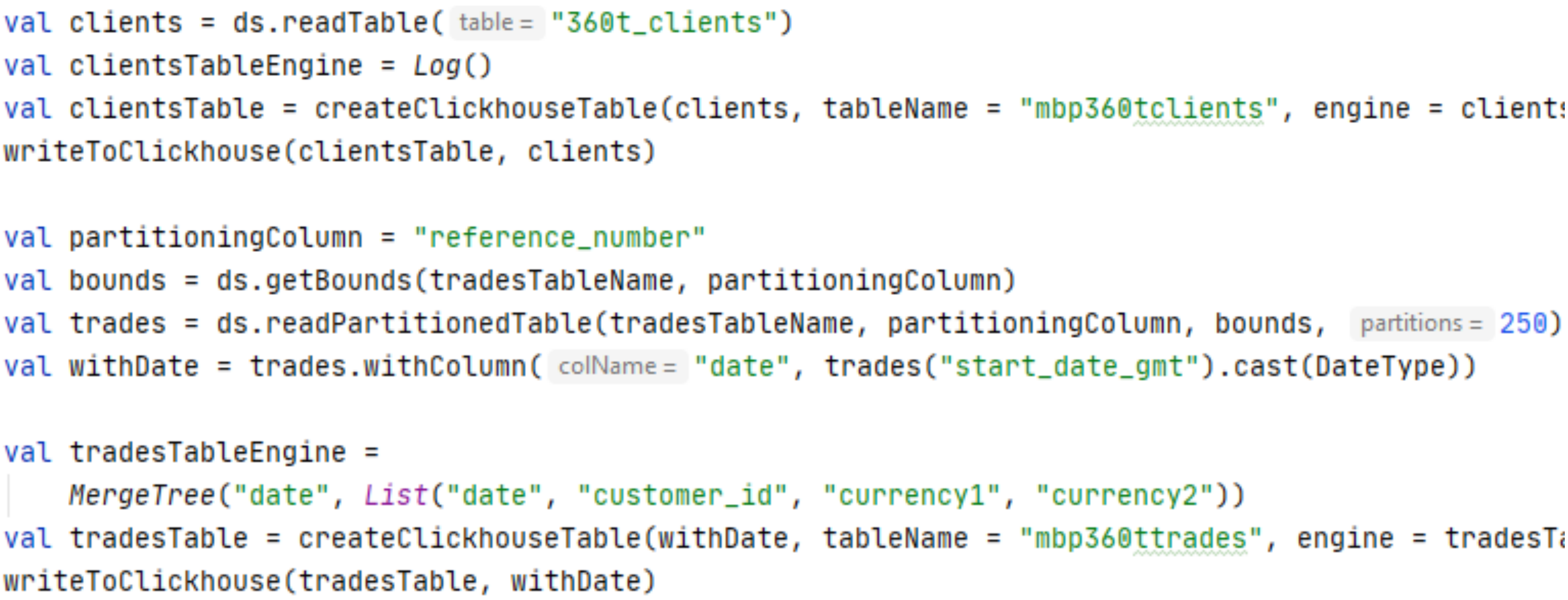
```
select salesperson, sum(profit)
 from tradedb.trades
  where (team in (
   select team from salesdb.sales
   where person = 'john.smith@db.com'
  ) and
   country = 'RU') or isInternal
 group by salesperson
```

https://github.com/yandex/ClickHouse/issues/523

Scalable DWH on Clickhouse

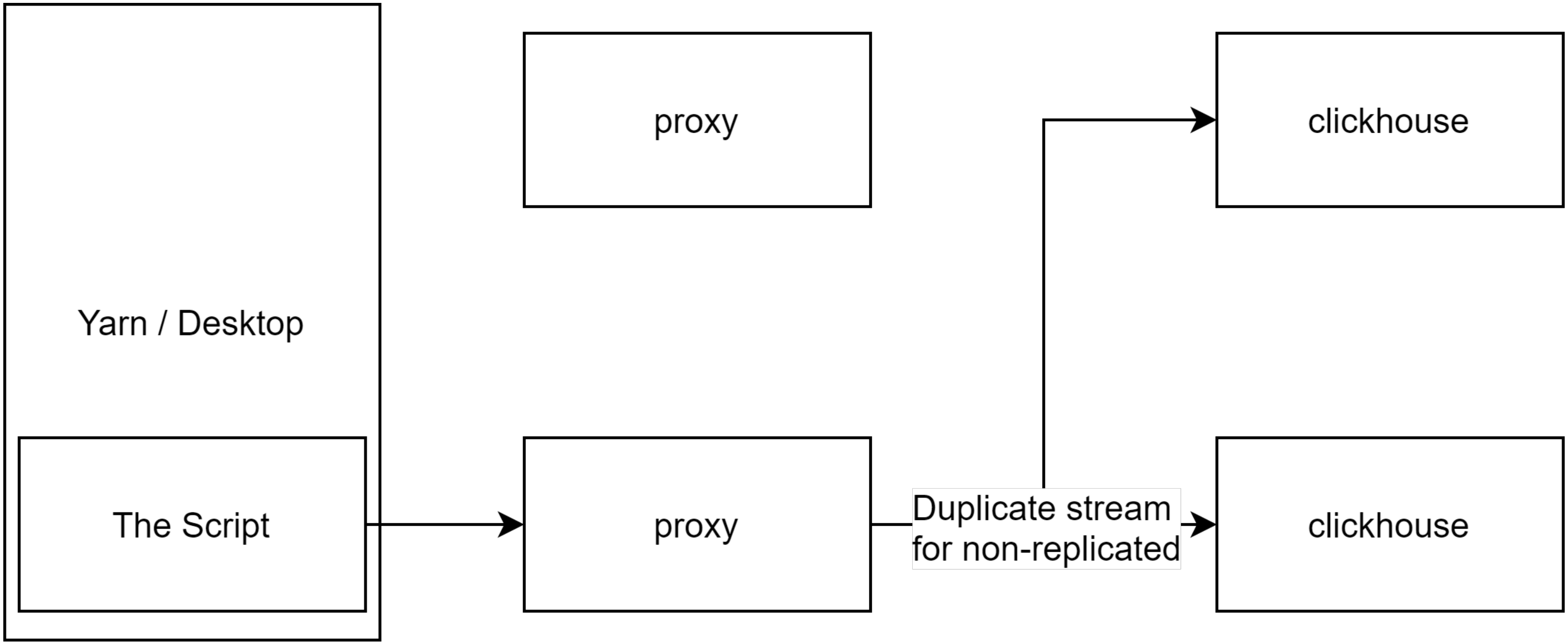


The ETLLib



```
val clients = ds.readTable( table = "360t_clients")
val clientsTableEngine = Log()
val clientsTable = createClickhouseTable(clients, tableName = "mbp360tclients", engine = client
writeToClickhouse(clientsTable, clients)

val partitioningColumn = "reference_number"
val bounds = ds.getBounds(tradesTableName, partitioningColumn)
val trades = ds.readPartitionedTable(tradesTableName, partitioningColumn, bounds,  partitions = 250)
val withDate = trades.withColumn( colName = "date", trades("start_date_gmt").cast(DateType))

val tradesTableEngine =
    MergeTree("date", List("date", "customer_id", "currency1", "currency2"))
val tradesTable = createClickhouseTable(withDate, tableName = "mbp360ttrades", engine = tradesTa
writeToClickhouse(tradesTable, withDate)
```

The ETLLib



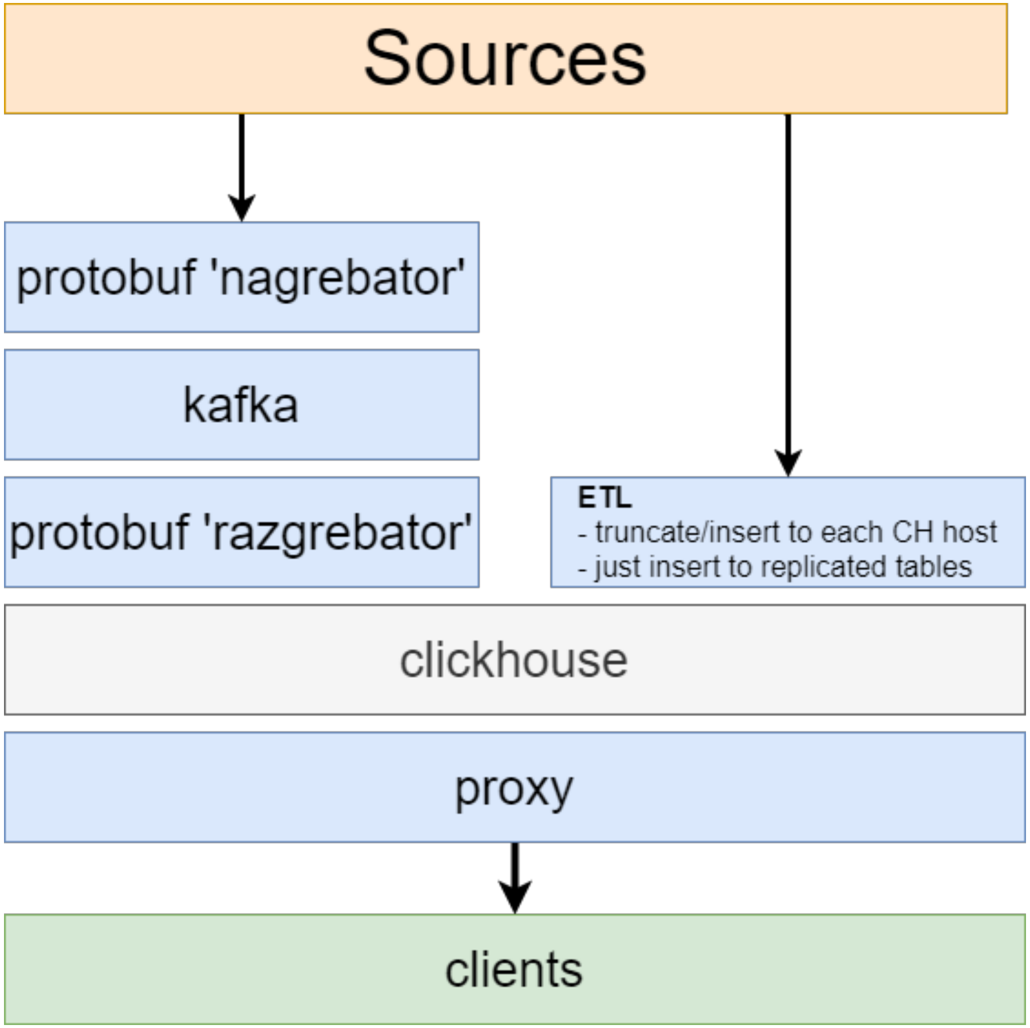
But then, an outage happens…

Consequences

- Extremely fast time to market
  - Datasource access -> report in 30 minutes, no effort from IT
  - we’ve closed a number of urgent reporting tasks with it
- Enourmous support cost.
  - >100 ETL jobs in a year after release
  - Unpredictable dependencies between the jobs
  - Unclear criticality
  - Lack of data quality
  - Substandard clickhouse performance, OOM killer, segfaults
  - Chronic outages

ETL rebirth

- No self-service
  - **Data Engineers** make the feeds
  - **Dataset** is the product
- Clickhouse is not an ETL engine!
  - All ETL is done with spark
- Proto+Kafka for external system feeds
  - But not Kafka Engine



The views curse

- View is an API!
  - But analysts do not know who depends on them and break stuff

- Solution
  - Only sandbox views for analysts, no access to he neighbor’s sandbox
  - No persistent reports from the sandbox
  - Common views under data engineer’s control

Current usage

- Tracking of changes/migrations/projects
- Investment planning
- Client interaction quality
- Research

Mostly small/medium heterogeneous data

- 72 datasources/systems
- 2 Rps
- 8 boxes 256-1500Gb RAM/26Tb
- 15Tb or 6Tb compressed largest table

Couple of epic wins

- Trading regulation
  - Delivered in 3 months, daily interactive analysis by risks department over 6TB (compressed, and growing) data
- Client regulation
  - In 1.5 months, holistic view on 5 trading systems activity, thanks to ETLLib
- ….

Thanks!

Disclaimer

Данный материал не является предложением или предоставлением какой-либо услуги. Данный материал предназначен исключительно для информационных и иллюстративных целей и не предназначен для распространения в рекламных целях. Любой анализ третьих сторон не предполагает какого-либо одобрения или рекомендации. Мнения, выраженные в данном материале, являются актуальными на текущий момент, появляются только в этом материале и могут быть изменены без предварительного уведомления. Эта информация предоставляется с пониманием того, что в отношении материала, предоставленного здесь, вы будете принимать самостоятельное решение в отношении любых действий в связи с настоящим материалом, и это решение является основанным на вашем собственном суждении, и что вы способны понять и оценить последствия этих действий. ООО "Дойче Банк ТехЦентр" не несет никакой ответственности за любые убытки любого рода, относящихся к этому материалу.