

Stambiaのご紹介

Stambiaの特徴

・対応力

- JAVAを利用
- 幅広いデータソースの対応

・効率的

- E-LTアーキテクチャを採用

・使いやすさ

- テンプレート機能で自動化
- データソースを一元的に操作可能

Stambiaの特徴

・対応力

- JAVAを利用
- 幅広いデータソースの対応

・効率的

- E-LTアーキテクチャを採用

・使いやすさ

- テンプレート機能で自動化
- データソースを一元的に操作可能

RDBMS

～対応力～

- Actian Ingres
- IBM DB2 UDB
V8.x, V9.x, V10.x
- IBM DB2/400, DB2/MVS
- Informix
- Oracle, Oracle Hexadata,
8i, 9i, 10g, 11g, 12c
- Oracle RDB
- Microsoft SQL Server,
2005, 2008, 2012, 2014
- MySQL,
V5.0 to 5.7 and
followings
- Postgre SQL, V8.x, V9.x
- SAP Sybase ASE

クラウド

～対応力～

- Amazon Redshift
- Amazon RDS
 - Oracle
 - MySQL
 - Amazon Aurora
 - MariaDB
 - Microsoft SQL Server
- Google Spreadsheet



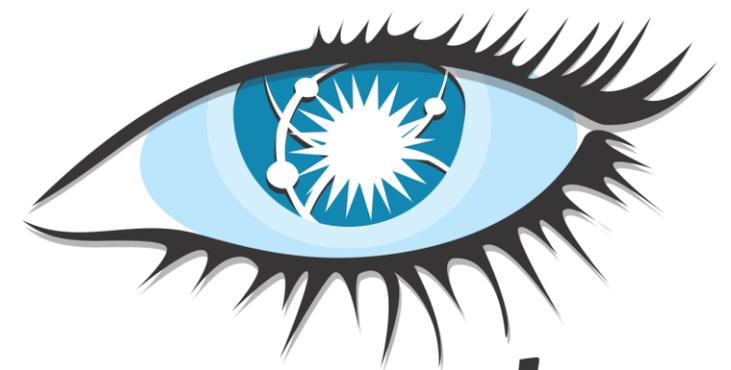
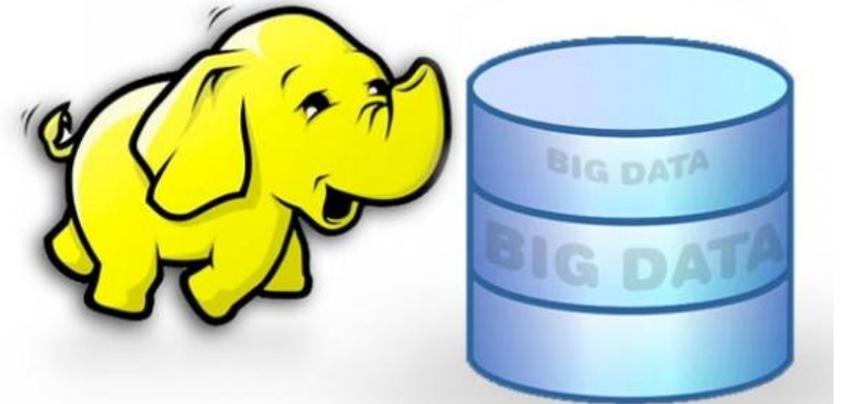
Hadoop / No SQL

～対応力～

- Hive
- Cassandra
- Elastic Search
- Parstream



elasticsearch



cassandra

その他

- DBase
- H2
- Hyperfile
- Hypersonic SQL
- Lotus Notes
- Microsoft Access
- Microsoft Excel
- Paradox
- Monet DB
- HP Vertica
- SAP Sybase IQ
- SAP Hana
- Salesforce
- Actian Vectorwise

～対応力～

- EMC Greenplum
- IBM Netezza (Pure Data)
- Teradata V13, V14, V15
- CSV, XML
- WEB API

一覽

- Actian Ingres
- IBM DB2 UDB V8.x, V9.x, V10.x
- IBM DB2/400, DB2/MVS
- Informix
- Oracle, Oracle Hexadata, 8i, 9i, 10g, 11g, 12c
- Oracle RDB
- Microsoft SQL Server, 2005, 2008, 2012, 2014
- MySQL, V5.0 to 5.7 and followings
- Postgre SQL, V8.x, V9.x
- Hyperfile
- SAP Sybase ASE
- Hive
- Cassandra
- Elastic Search
- Parstream
- SAP Sybase IQ
- SAP Hana
- Monet DB
- Amazon Redshift
- Amazon RDS
- Google Spreadsheet
- DBase

～対応力～

- Hypersonic SQL
- Lotus Notes
- Microsoft Access
- Microsoft Excel
- Paradox
- EMC Greenplum
- IBM Netezza (Pure Data)
- Teradata V13, V14, V15
- Actian Vectorwise
- HP Vertica
- H2

Stambiaの特徴

・対応力

- JAVAを利用
- 幅広いデータソースの対応

・効率的

- E-LTアーキテクチャを採用

・使いやすさ

- テンプレート機能で自動化
- データソースを一元的に操作可能

豊富なテンプレート

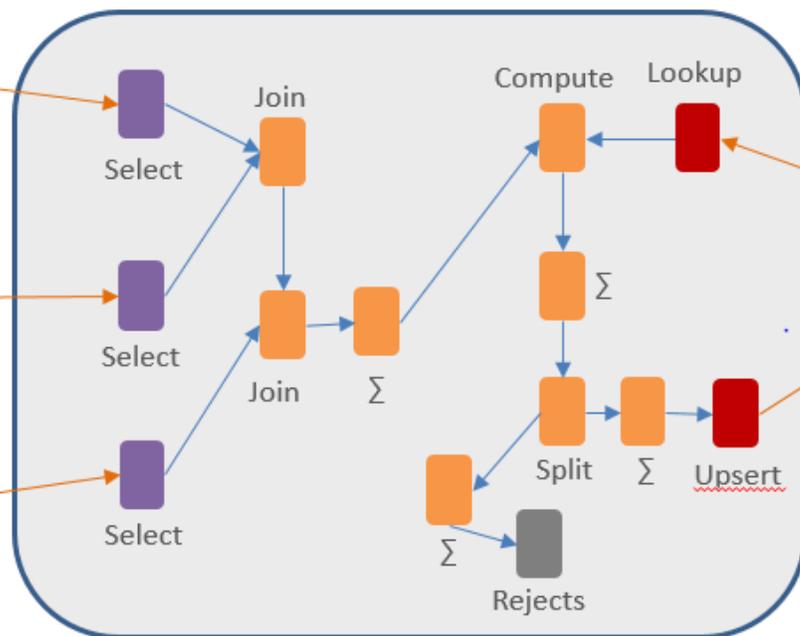
～使いやすさ～

不慣れな方でも、簡単に開発可能

(select, transformation, split, lookup, join, など)



ソース側



ターゲット側

簡潔な開発

～使いやすさ～

データの関連付け、フィルタ、変換などを
マッピング機能によりグラフィカルに設定が可能



ソース側

T_BILLING_LINES	
I	BLL_ID
I	BIL_ID
D	BLL_QTY
D	BLL_DISCOUNT_RATE
N	BLL_DISCOUNT_AMOUNT
D	BLL_AMOUNT
V	BLL_TYPE
I	BDR_ID

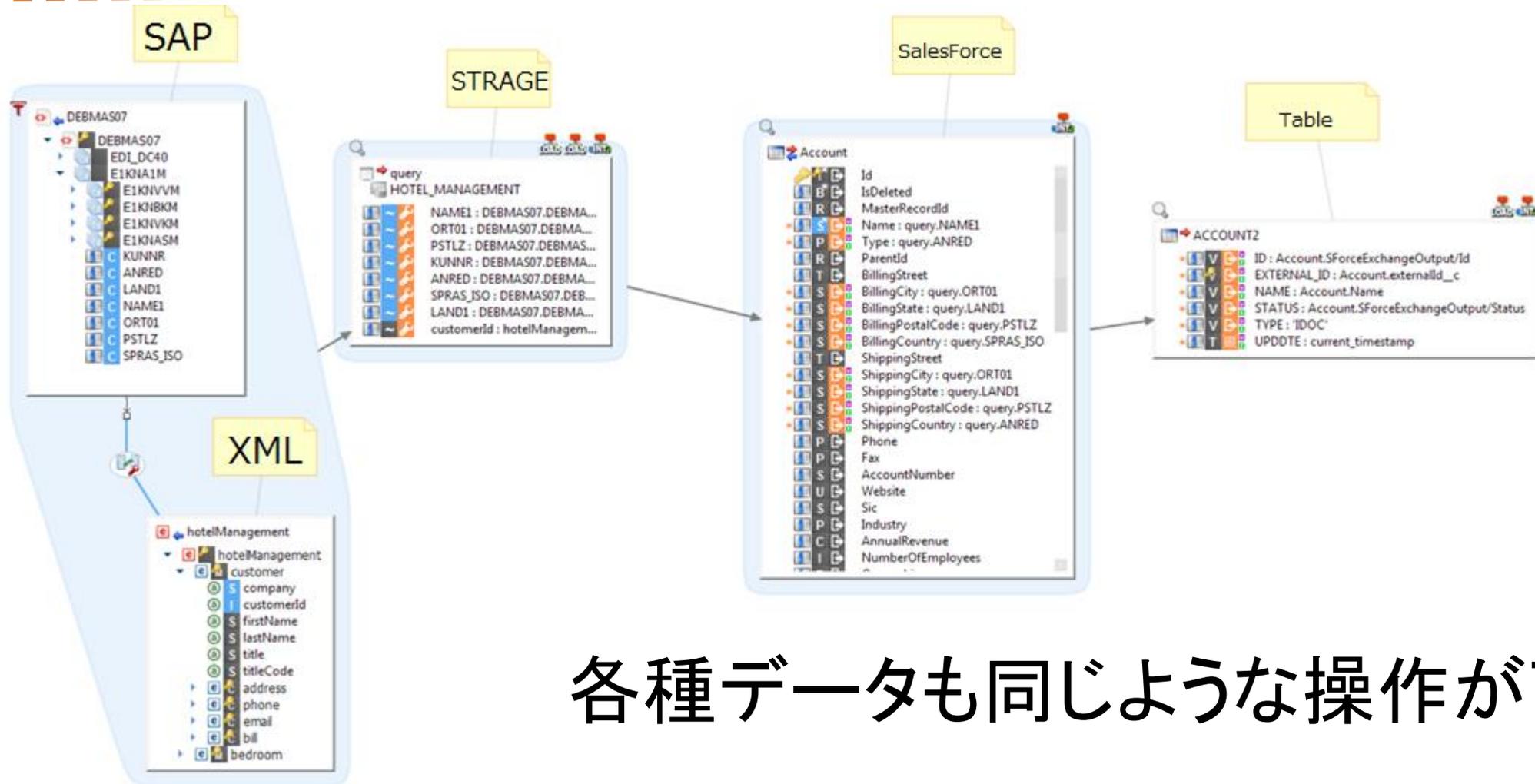
T_BILLING	
I	BIL_ID
I	CUS_ID
C	PMT_CODE
T	BIL_DATE
T	BIL_PMT_DATE

FACT_BILLING	
I	BIL_KEY_ID <-
I	BIL_ID <- T_BILLING.BIL_ID
I	CUS_ID <- T_BILLING.CUS_ID
V	TIME_KEY_DAY <- DIM_TIME.TIME_K...
V	DIS_RANGE <- DIM_DISCOUNT.DIS_R...
V	PMT_CODE <- T_BILLING.PMT_CO...
N	BIL_AMOUNT <- SUM(T_BILLING_...
N	BIL_QTY <- SUM(T_BILLING_LINE...
T	UPDATE_DATE <- current_timestamp
	PK_FACT_BILLING (pk) : false
	FK_FACT_BILLING_CUS (fk) : true
	FK_FACT_BILLING_DIS (fk) : true
	FK_FACT_BILLING_PMT (fk) : true
	FK_FACT_BILLING_TIME (fk) : true
	CK_BIL_AMOUNT (ck) : true



ターゲット側

ユニバーサルマッピング ~使いやすさ~



各種データも同じような操作が可能

Stambiaの特徴

・対応力

- JAVAを利用
- 幅広いデータソースの対応

・使いやすさ

- テンプレート機能で自動化
- データソースを一元的に操作可能

・効率的

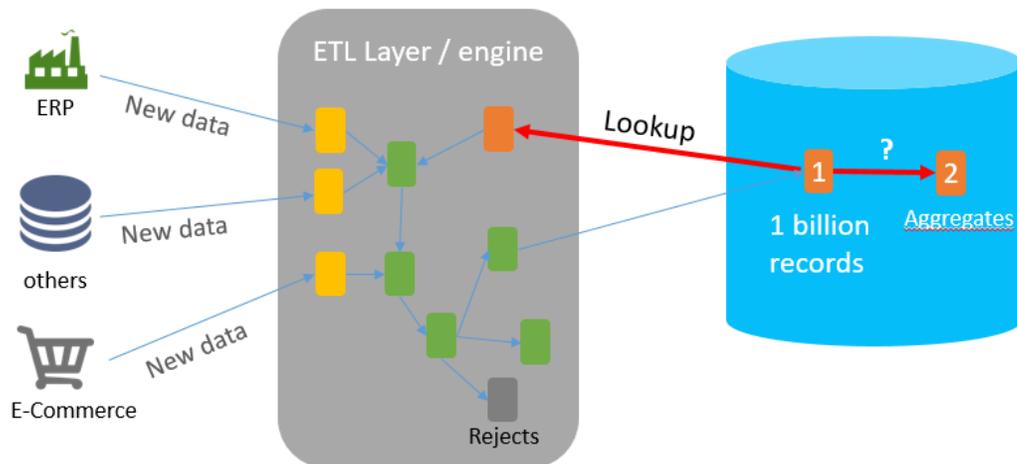
- E-LTアーキテクチャを採用

E-LT アーキテクチャ

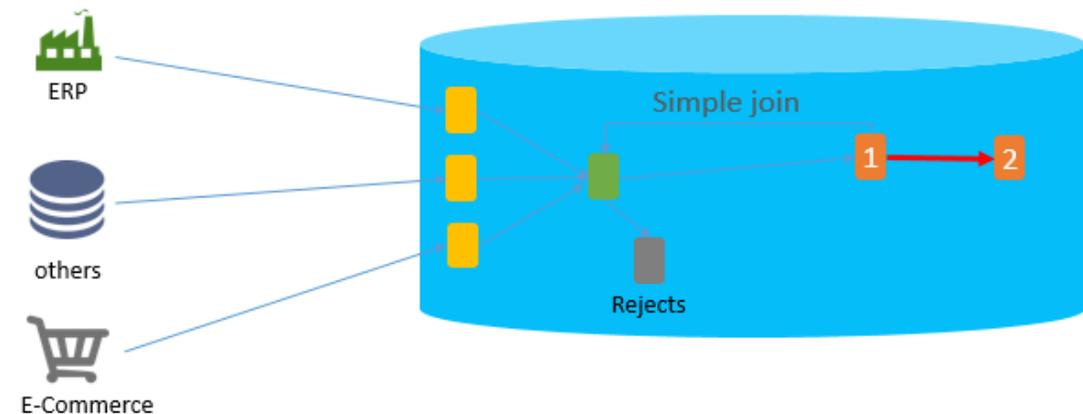
～効率的～

ETLツール側が行っていたデータの変換処理を
データストア側に任せることでボトルネックを回避

従来アーキテクチャ



E-LTアーキテクチャ



Stambiaの特徴

・対応力

- JAVAを利用
- 幅広いデータソースの対応

・効率的

- E-LTアーキテクチャを採用

・使いやすさ

- テンプレート機能で自動化
- データソースを一元的に操作可能

Stambiaの特徴

- ・対応力 ▶ **高速**に対応

- JAVAを利用

- 幅広いデータソースの対応

- ・効率的 ▶ **高速**に処理

- E-LTアーキテクチャを採用

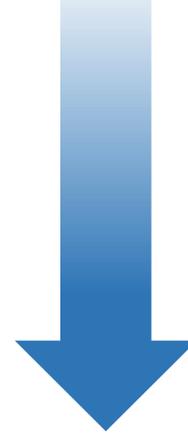
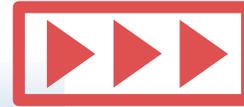
- ・使いやすさ ▶ **高速**に開発

- テンプレート機能で自動化

- データソースを一元的に操作可能

まとめ

- ・対応力 ▶ 高速に対応
- ・使いやすさ ▶ 高速に開発
- ・効率的 ▶ 高速に処理



開発から実行まで
高速にデータ統合が可能