Asakusa Framework適用判断ポイント

OSSコンソーシアム　Asakusa Framework部会
http://www.osscons.jp/asakusafw/

はじめに

目的及び対象

OSS Consortium

- 本資料の目的
  - 本資料は、Asakusa Frameworkの適用を検討する初期段階で、適用可否や向き不向きの判断を支援するための資料です。
  - 本資料は、以下の情報を提供します。
    - 適用可否を判断する上での判断ポイント
    - 適用効果(性能面)及び適用効率(コスト)で留意すべき事項

- 本資料の対象
  - 本資料が想定するAsakusa Frameworkは以下です。
    - Asakusa Framework （0.8.0以降）
    - Asakusa on Spark （0.3.0以降）

    （注） Asakusa on M^{3}BP （0.1.0以降)は想定していません。
    ただし、当てはまる部分もあるので、参考にすることは可能です。
  - 本資料が想定する開発対象は以下です。
    - 高速な処理が望まれるバッチ処理の新規開発
    - 既存バッチ高速化のためのマイグレーション

対象者及び前提知識

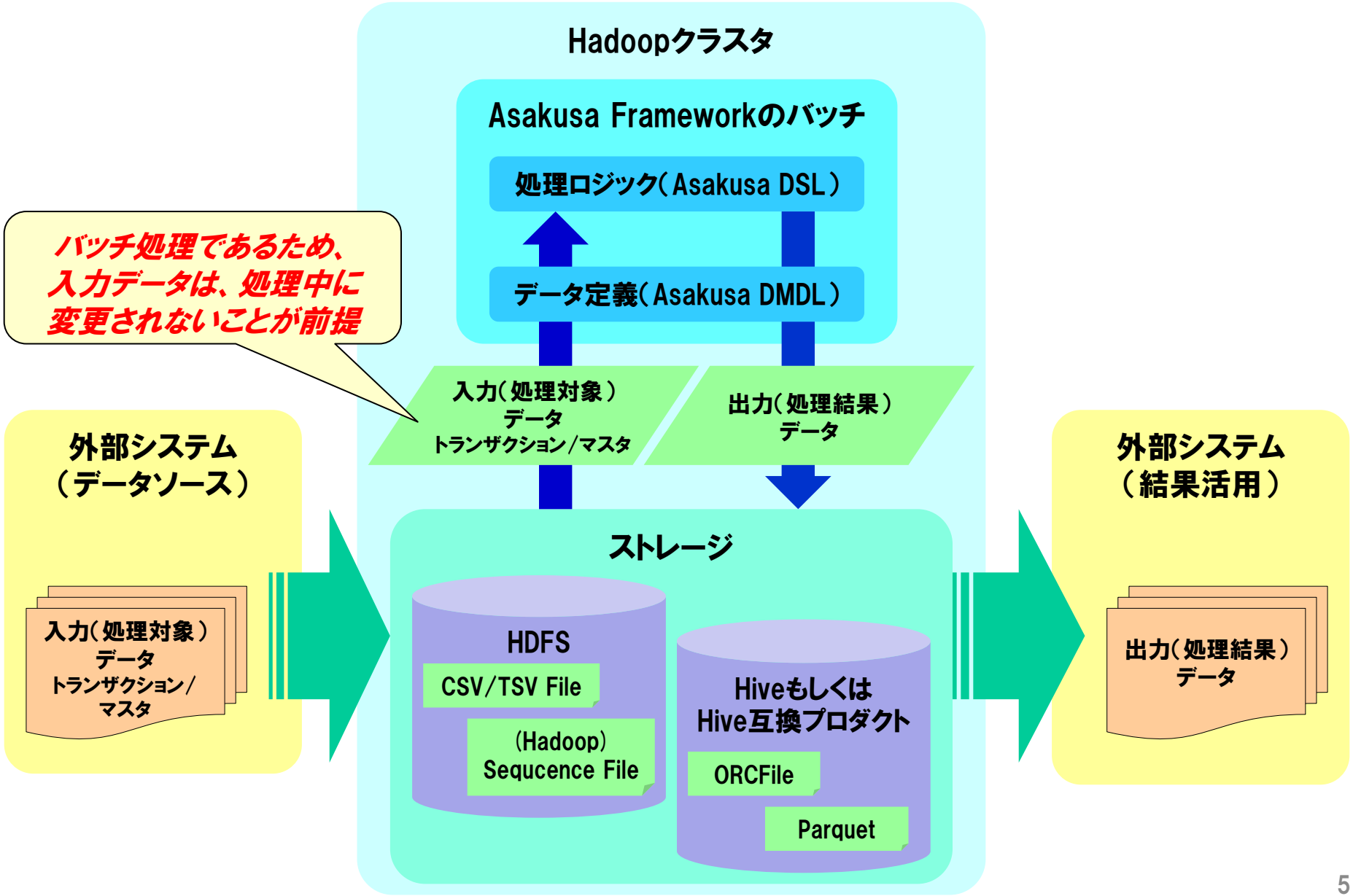
OSS Consortium

- 本資料の対象者
  - Asakusa Frameworkの適用を検討している方
  - Asakusa Frameworkを適用したシステムを提案する方
  - 以下を検討もしくは提案を行う方
    - 高速な処理が望まれるバッチ処理の新規開発
    - 既存バッチ高速化のためのマイグレーション

- 必要とされる前提知識
  - Asakusa Frameworkの概要
    http://docs.asakusafw.com/latest/release/ja/html/introduction/overview.html
  - Asakusa Data Model（Asakusa DMDL）
    http://docs.asakusafw.com/latest/release/ja/html/dmdl/index.html
  - Asakusa DSL
    http://docs.asakusafw.com/latest/release/ja/html/dsl/index.html
  - 外部システム連携ツール(Direct I/O、WindGate）
    http://docs.asakusafw.com/latest/release/ja/html/directio/index.html
    http://docs.asakusafw.com/latest/release/ja/html/windgate/index.html

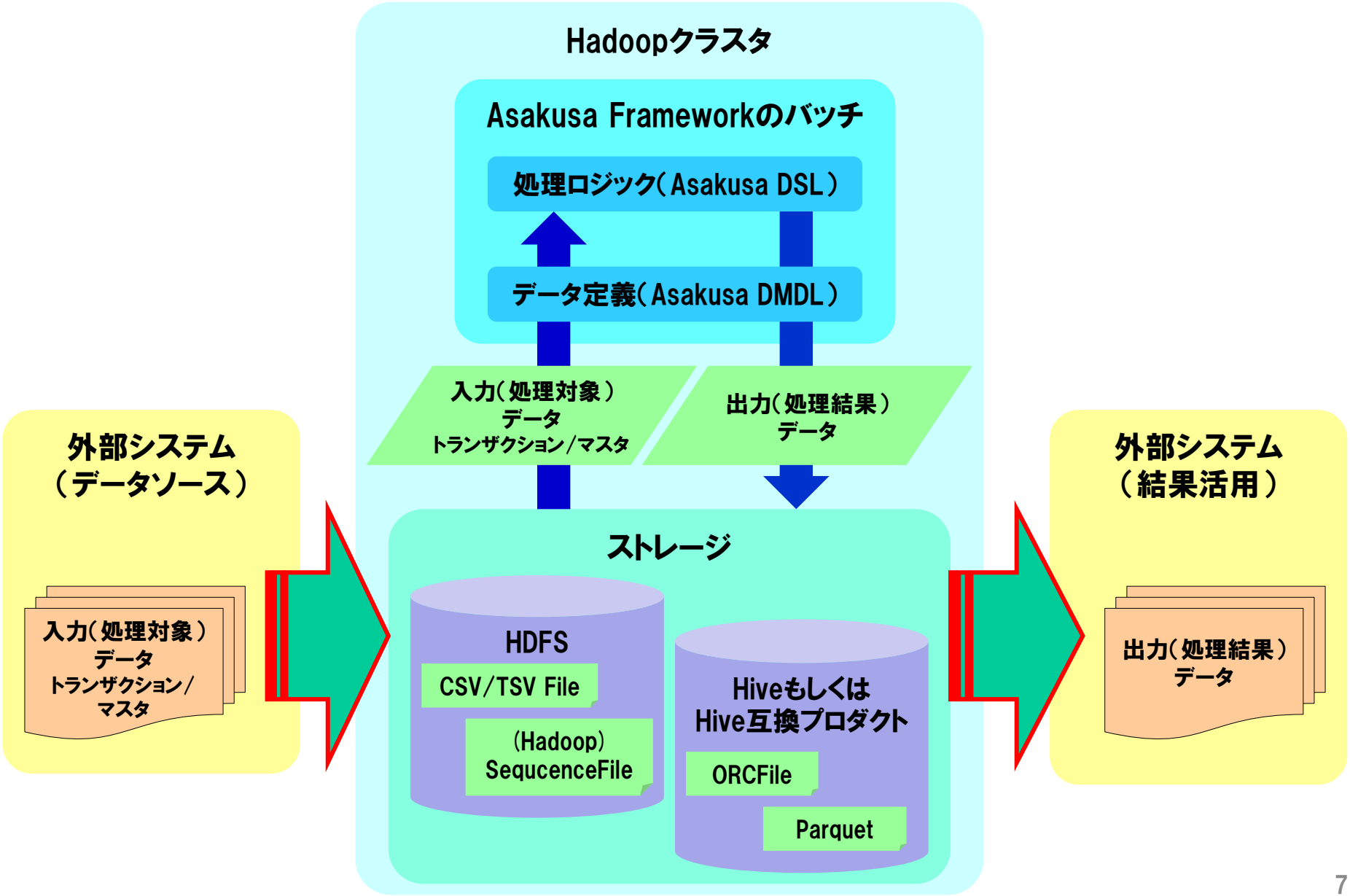
Asakusa Frameworkを適用したシステムの全体像

Asakusa Frameworkのバッチ処理概要

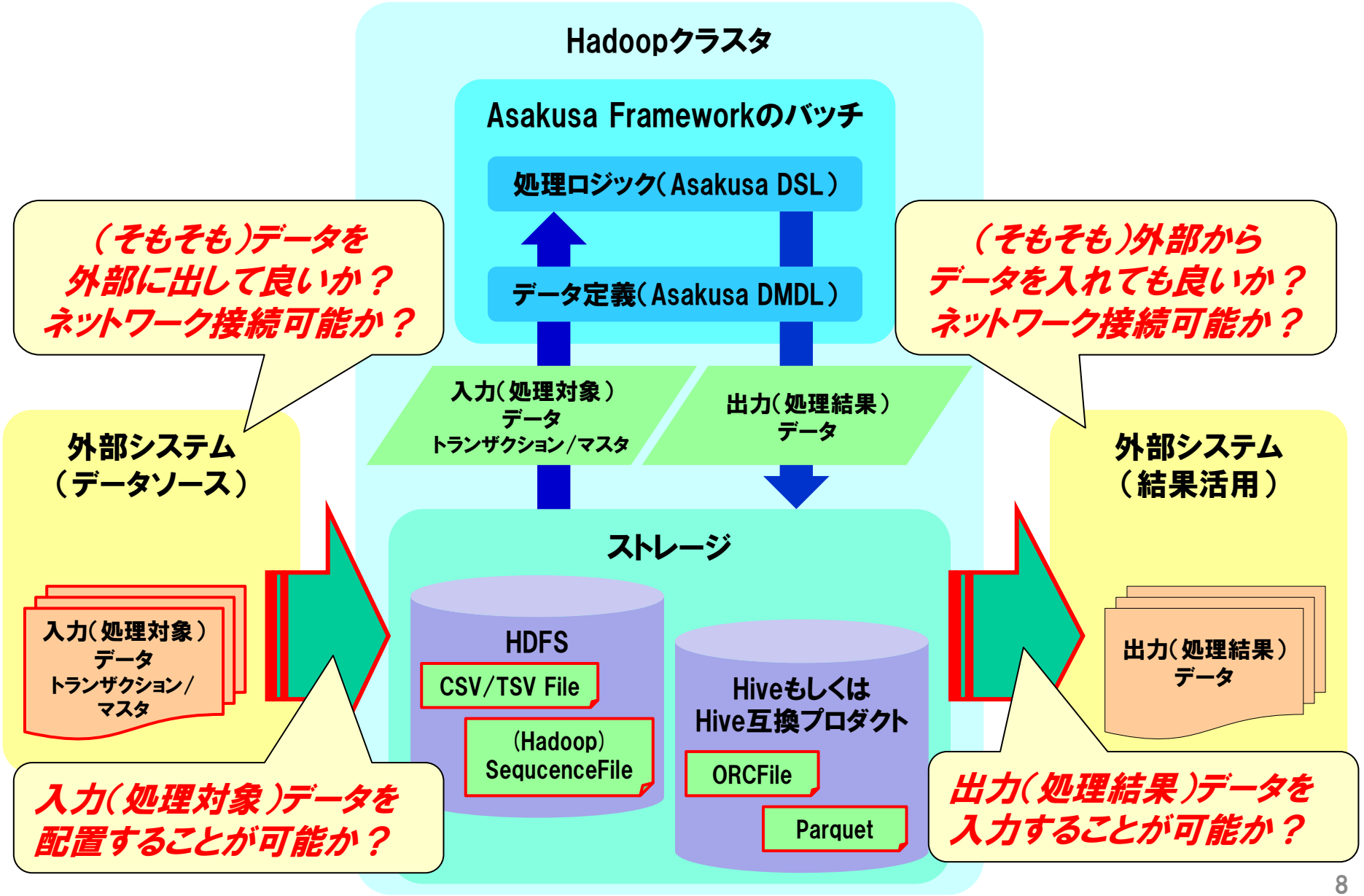


適用判断ポイント（１）
データ連携

データ連携


Hadoopクラスタ
Asakusa Frameworkのバッチ
処理ロジック（Asakusa DSL）
データ定義（Asakusa DMDL）
入力（処理対象）
データ
トランザクション/マスタ
出力（処理結果）
データ
外部システム
（データソース）
入力（処理対象）
データ
トランザクション/
マスタ
ストレージ
HDFS
CSV/TSV File
（Hadoop）
SequcenceFile
Hiveもしくは
Hive互換プロダクト
ORCFile
Parquet
外部システム
（結果活用）
出力（処理結果）
データ

データ連携の判断ポイント(１)



データ連携の判断ポイント(２)

OSS Consortium

- 外部システムが、以下を直接利用する/利用可能な場合は問題なし
  - HDFS上のCSV/TSV形式のファイル
  - HDFS上のHadoop SequenceFile形式のファイル
  - HiveもしくはHive互換プロダクト上のORCFile形式、Parquet形式のファイル

- 外部システムが、以下の場合は基本的に問題なし
  - RDB上のテーブルの場合
    - 標準SQL/JDBCインターフェースで標準ツール(WindGate等)により連携
    - CSV/TSV形式のファイルでImport/Exportし、HDFSに出し入れ
  - CSV/TSV形式のファイル等、Import/Export可能な場合
    - 直接HDFSに出し入れ
    - 標準ツール(WindGate等)で連携

データ連携の判断ポイント(３)

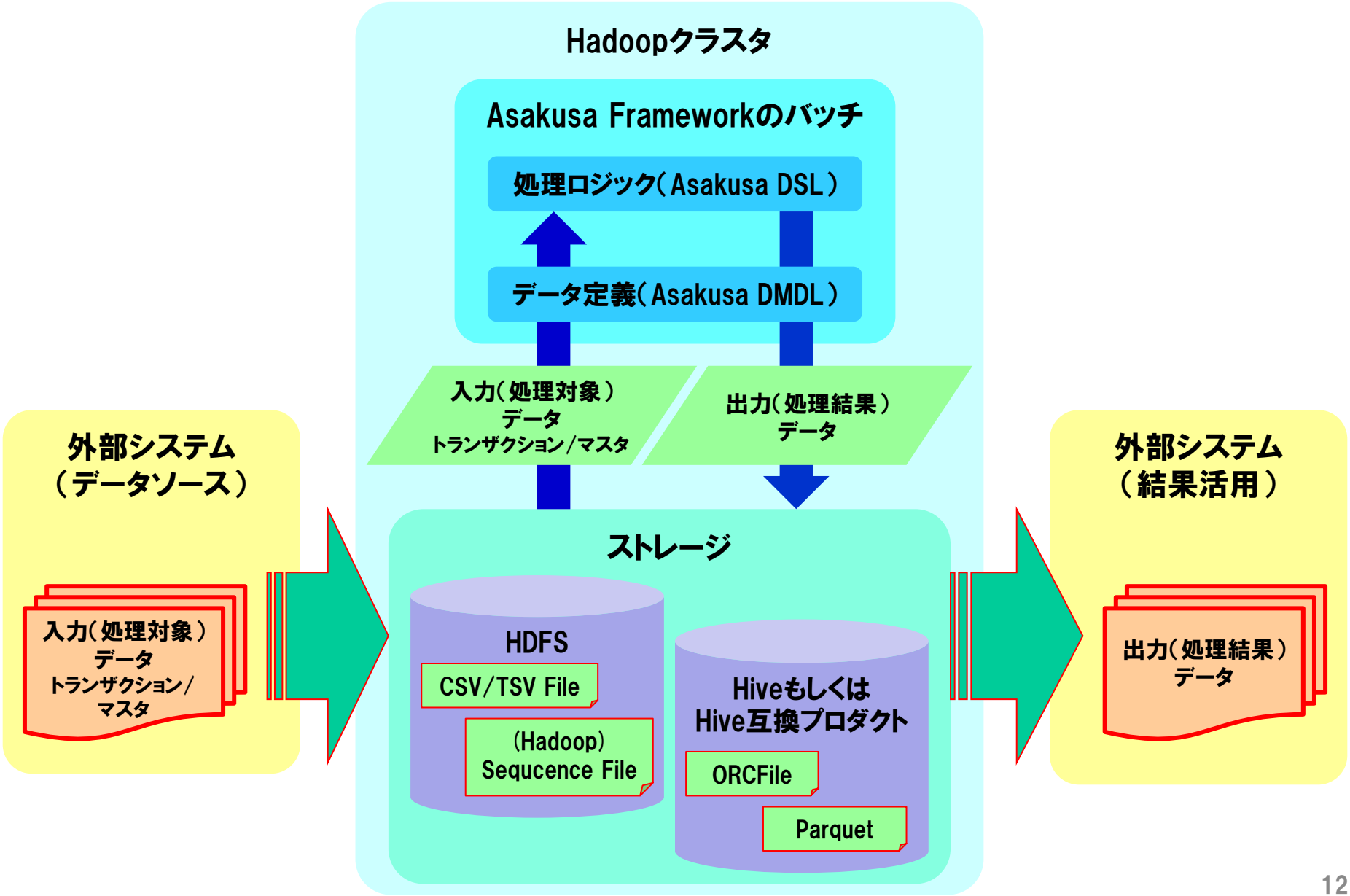
OSS Consortium

- その他の場合は要検討
  - 一般的にはCSV/TSV形式のファイルで連携する
  - 外部システムでCSV/TSV形式のファイルをImport/Exportする手段を検討
    - 標準機能や導入済み製品の機能を利用
    - データ連携製品等を新規導入
    - Import/Export用アドインを新規開発

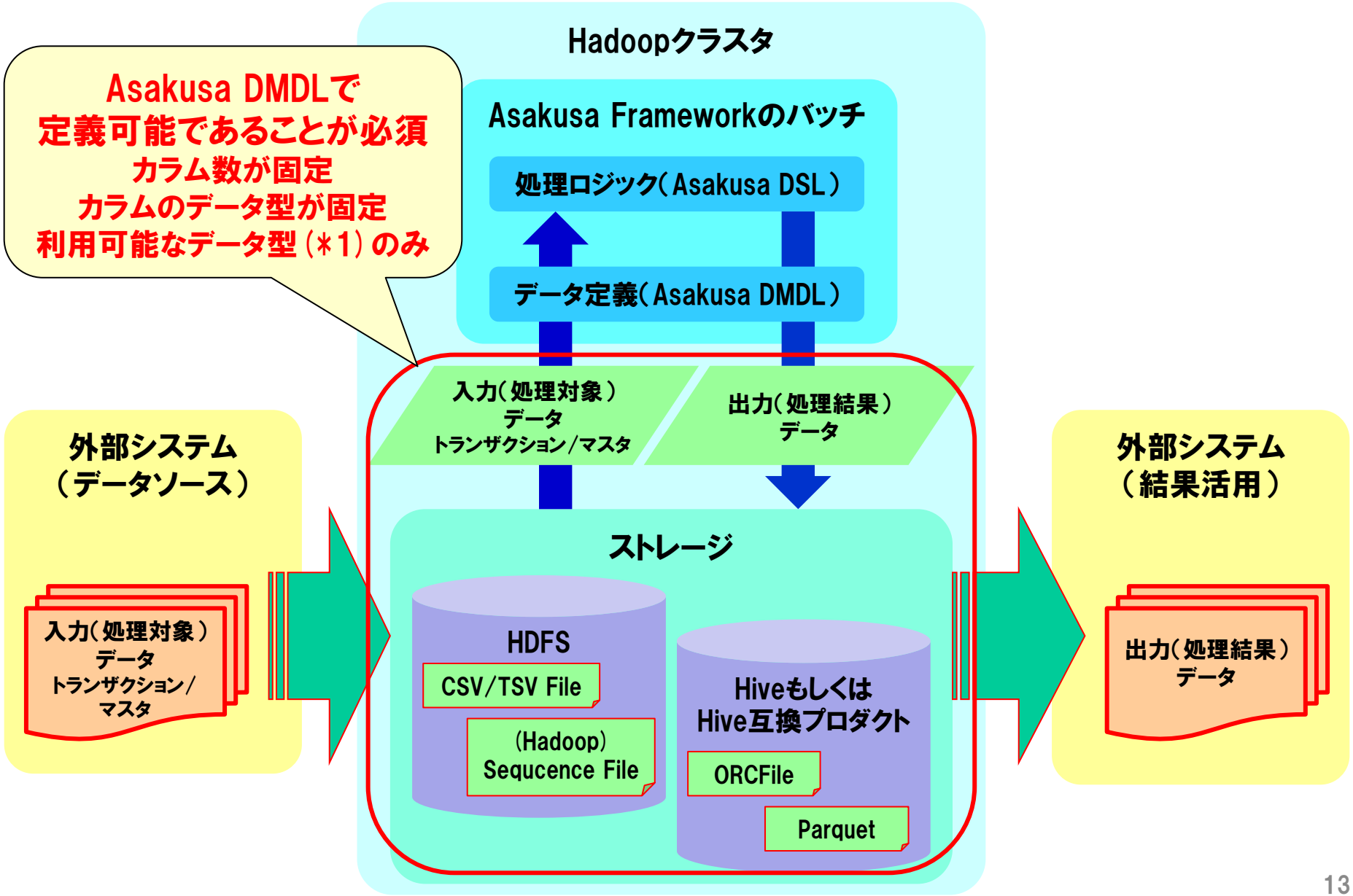
    ・・・

適用判断ポイント（２）
入出力データ

入出力データ


Hadoopクラスタ
Asakusa Frameworkのバッチ
処理ロジック（Asakusa DSL）
データ定義（Asakusa DMDL）
入力（処理対象）
データ
トランザクション/マスタ
出力（処理結果）
データ
外部システム
（データソース）
入力（処理対象）
データ
トランザクション/
マスタ
ストレージ
HDFS
CSV/TSV File
（Hadoop）
Sequence File
Hiveもしくは
Hive互換プロダクト
ORCFile
Parquet
外部システム
（結果活用）
出力（処理結果）
データ

入出力データの判断ポイント(１)



(*1) Asakusa DMDLで利用可能なデータ型

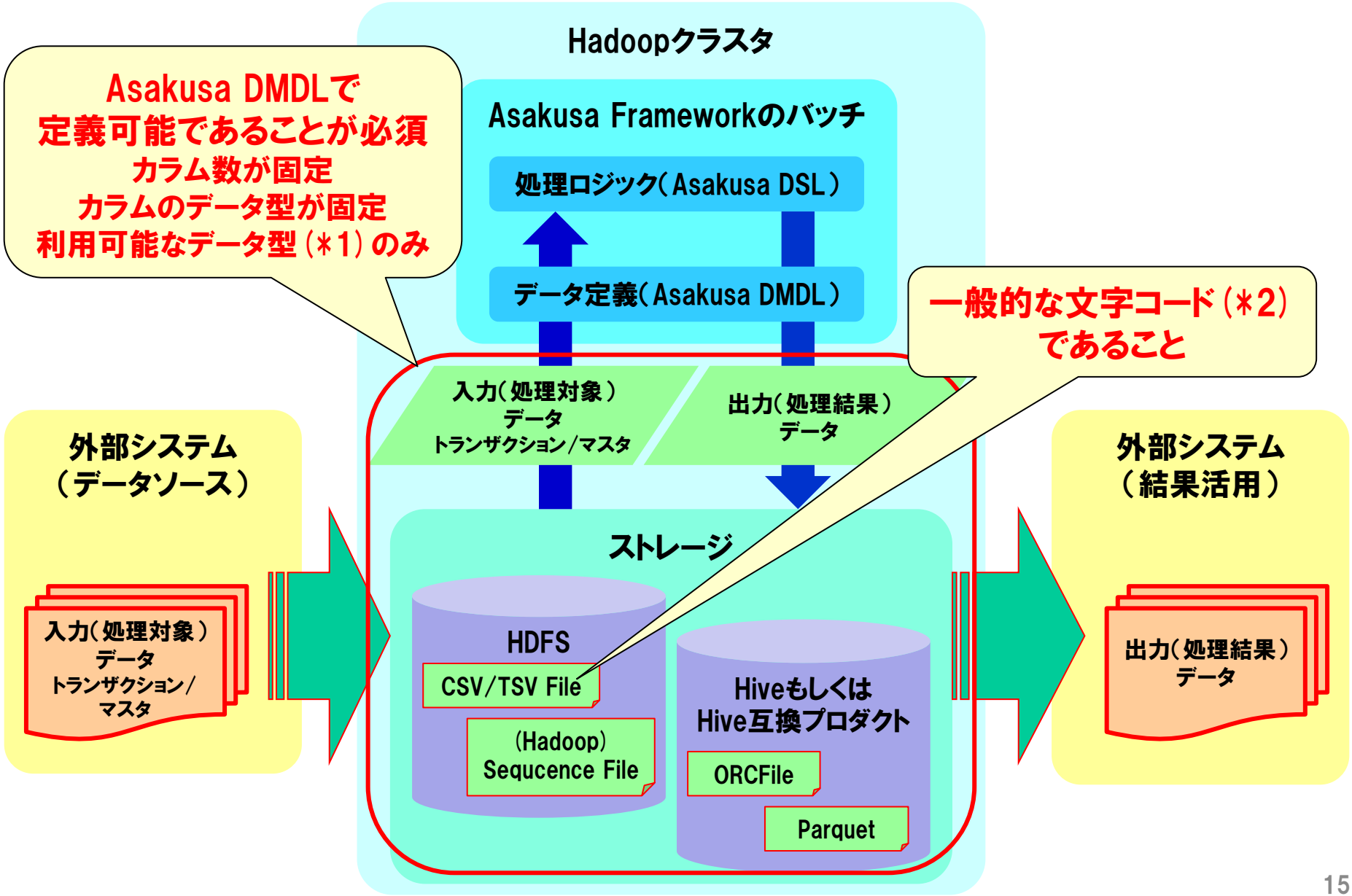
OSS Consortium

DMDLとJavaとJDBCのデータ型

説明	DMDL	Javaクラス	JDBC
32bit符号付き整数	INT	int（IntOption）	int
64bit符号付き整数	LONG	long（LongOption）	long
単精度浮動小数点	FLOAT	float（FloatOption）	float
倍精度浮動小数点	DOUBLE	double（DoubleOption）	double
文字列	TEXT	Text（StringOption）	String
10進数	DECIMAL	BigDecimal（DecimalOption）	BigDecimal
日付	DATE	Date（DateOption）	java.sql.Date
日時	DATETIME	DateTime（DateTime）	java.sql.Timestamp
論理値	BOOLEAN	boolean（BooleanOption）	boolean
8bit符号付き整数	BYTE	byte（ByteOption）	byte
16bit符号付き整数	SHORT	short（ShortOption）	short

http://docs.asakusafw.com/latest/release/ja/html/windgate/user-guide.html#dmdljdbc

入出力データの判断ポイント(２)



(*2) 一般的な文字コードとは？

OSS Consortium

- 入出力で使われる文字コードの既定値は「UTF-8」
  - 変換が可能であれば「UTF-8」にしておくのが無難
    - http://docs.asakusafw.com/latest/release/ja/html/sandbox/directio-tsv.html#id2
    - http://docs.asakusafw.com/latest/release/ja/html/directio/csv-format.html#csv
- Linux/Javaで扱える文字コードであれば利用可能
  - ISO-2022-JPなど
    - https://docs.oracle.com/javase/jp/7/technotes/guides/intl/encoding.doc.html
- メインフレームの場合は注意が必要
  - EBCDIC／EBCDIKなど、Linux/Javaで扱えない文字コードは、変換する手段を検討
    - 特に、日本語(カナ、ひらがな、漢字)や外字

入出力データの判断ポイント(３)

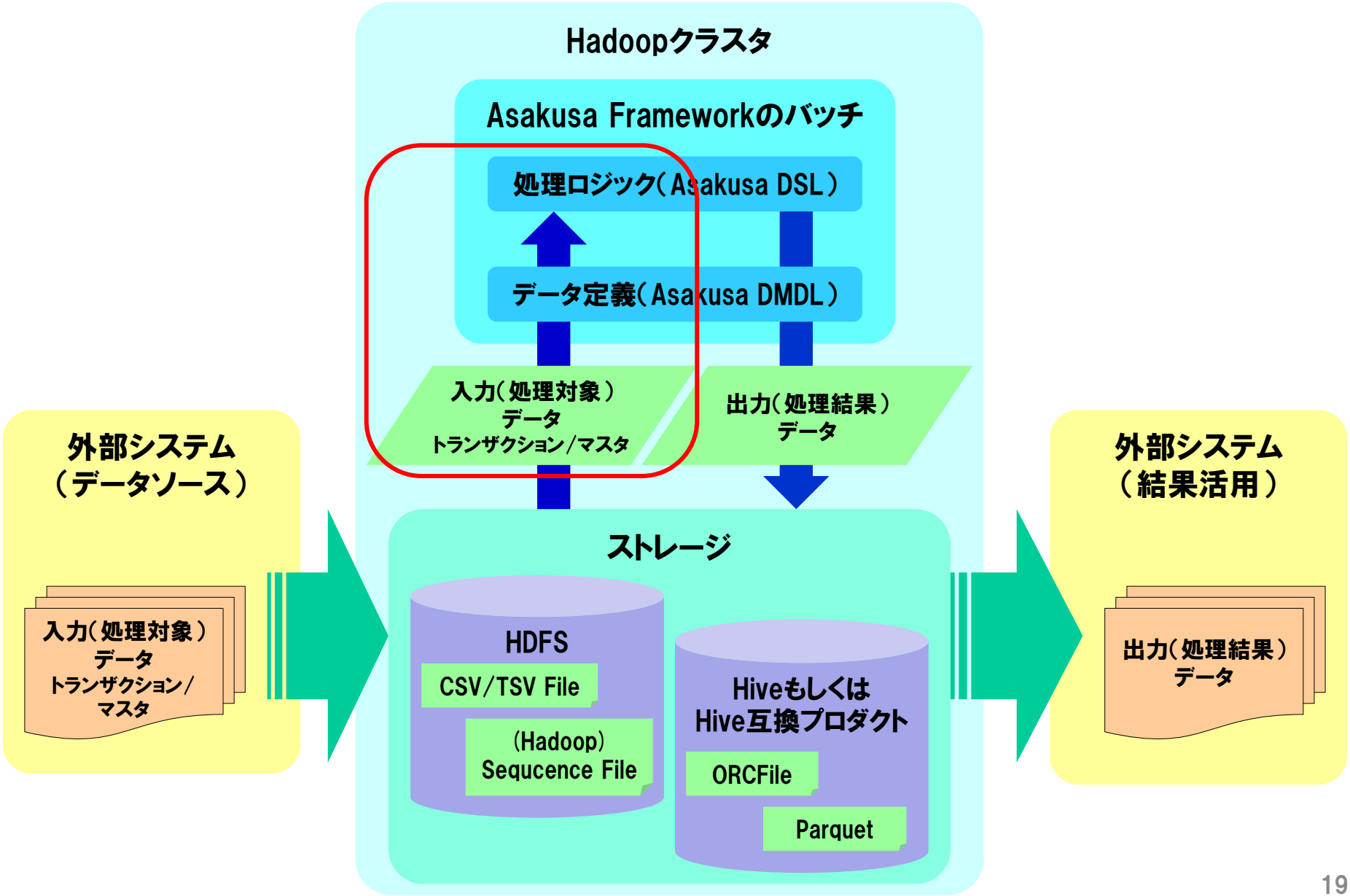
OSS Consortium

- RDB上のテーブルの場合は、大抵問題無い
  - DDLにより、カラム数やカラムのデータ型が決まっている
  - ただし、特殊な文字コードやデータ型を使っていないかの確認は必要
- RDB上のテーブル以外の場合は要確認
  - 条件に合う入出力データか？
    - カラム数は固定か？
    - カラムのデータ型は決まっているか？
    - 利用可能なデータ型(＊1)のみか？
    - 一般的な文字コード(＊2)であるか？
  - 条件に合わせたデータに変換することは可能か？手段はあるか？
    - 標準機能や導入済みの製品等で対応可能か？
    - データ連携製品等で可能か？新規導入は可能か？

    ・・・
- 条件に合わせることができなければ、適用は困難

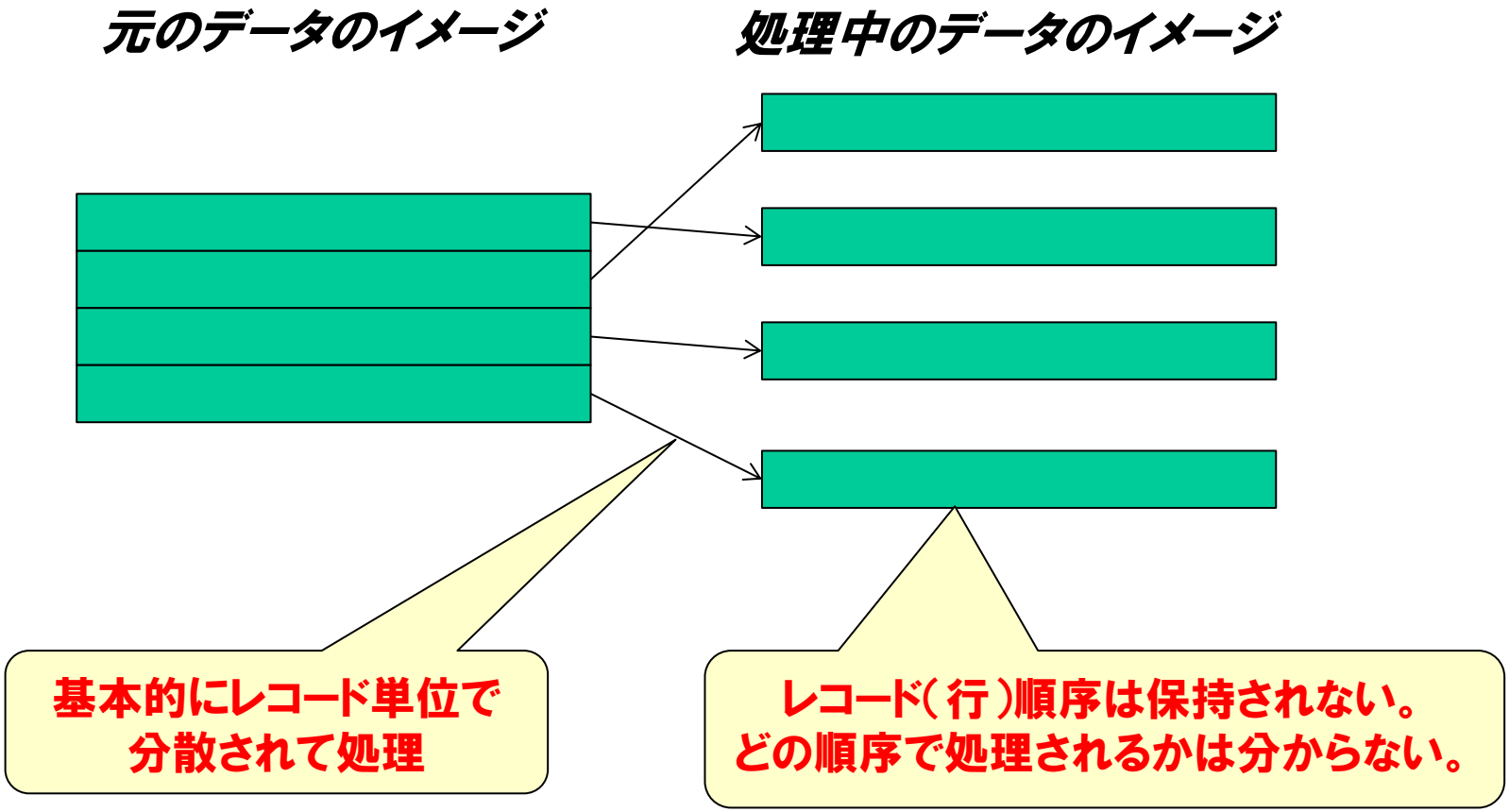
適用判断ポイント（３）
データの処理単位/データ順序依存性

Asakusa Frameworkのバッチ処理概要



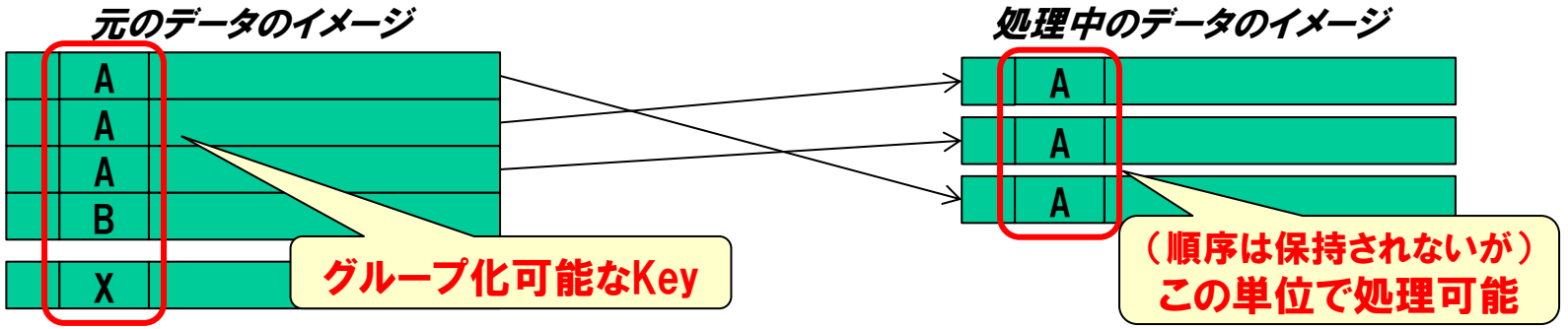
処理単位・処理順序の基本

- 基本的にはレコード（行）単位で処理される
- 元のレコード（行）順序は保持されない

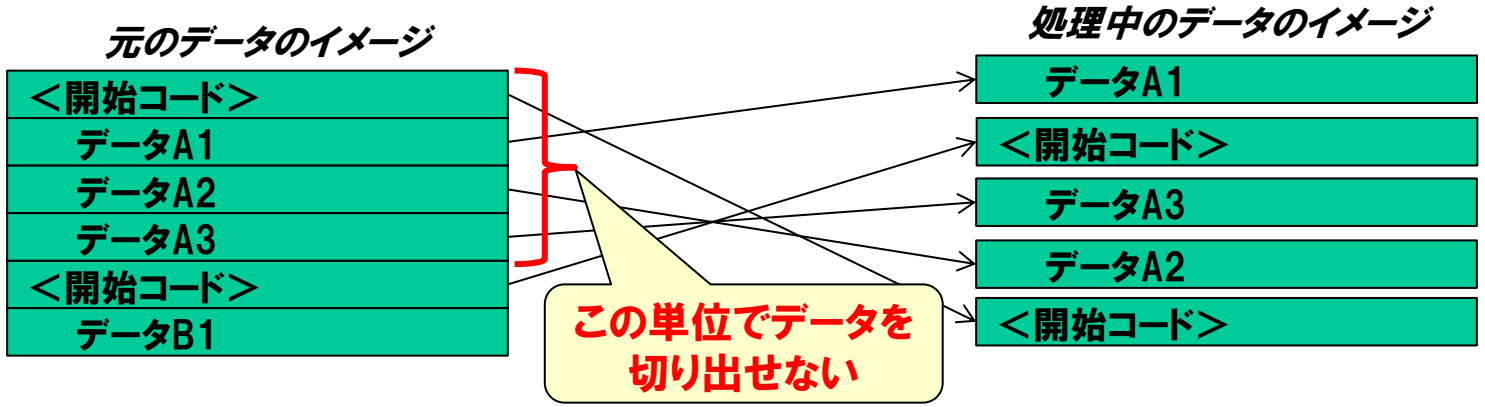


処理単位の判断ポイント

- 複数レコード(行)単位の処理が必要な場合
  - 処理単位でグループ化可能なKeyに相当するデータがあれば可能
    - グループ化可能なKeyがある例



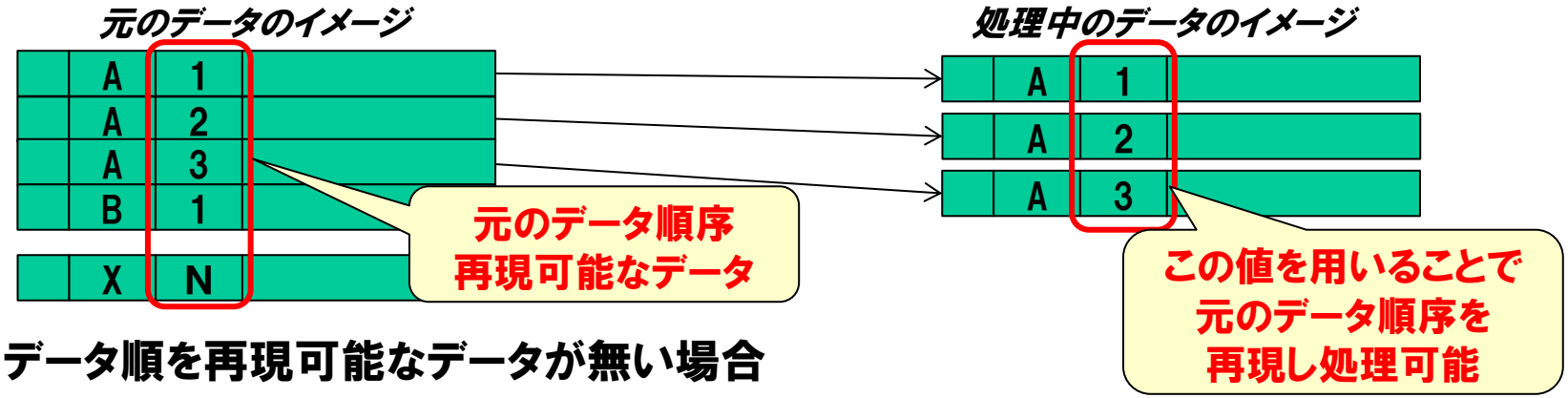
    - グループ化可能なKeyが無い例



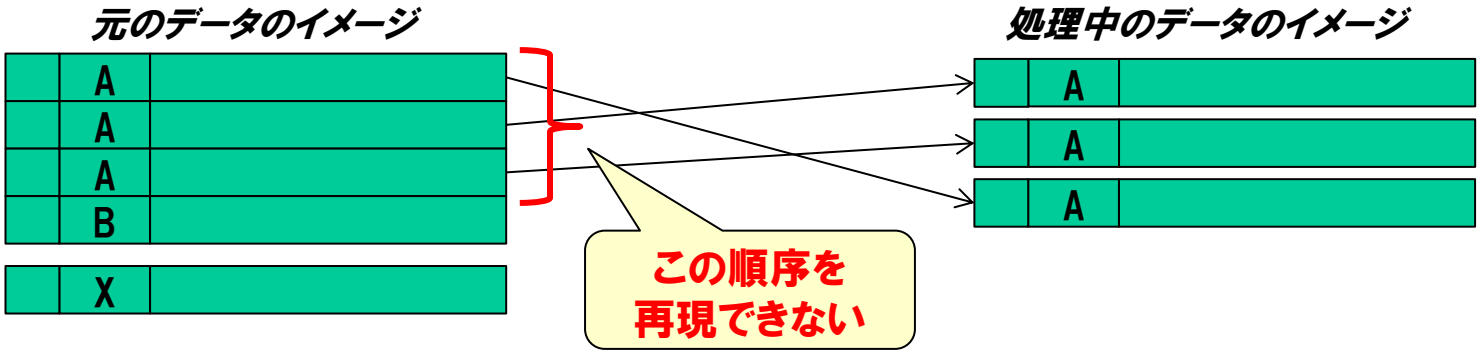
  - 元のデータになければ、グループ化を可能とするKeyが付与可能か検討

処理順序の判断ポイント

- （グループ化されたデータに対し）元のデータ順序に処理したい場合
  - データ順序を再現可能なデータがあれば可能
    - データ順を再現可能なデータがある場合



    - データ順を再現可能なデータが無い場合



  - 元のデータになければ、データ順序を再現可能とするデータを付与可能か検討

適用判断ポイント（４）
その他

その他の判断ポイント（１）

- 既存バッチ高速化におけるデータの整合性
  - 外部での処理を想定していない場合、思わぬ副作用が出る可能性がありうる。
  - データの整合性が保たれるかを要確認

- データ連携及びデータ整形・変換の実現コスト
  - 特にERPやメインフレームなどの場合は要注意
    - Import/Export用アドインの新規開発が必要な可能性が高い

- データ連携及びデータ整形・変換の時間的コスト
  - 以下の処理にかかる時間的コストを指す
    - 前述の適用判断ポイント「データ連携」のImport/Export処理
    - 前述の適用判断ポイント「データ連携」における、必要なデータ整形・変換処理
  - これらの時間的コストは、Asakusa Frameworkでは短縮できないため、
    利用するデータ連携ツールやネットワーク帯域などの外部環境も含めて判断が必要

その他の判断ポイント（２）

OSS Consortium

- メインフレーム等の固定長ファイルの扱い
  - 基本的には、「データ連携の判断ポイント」「入出力データの判断ポイント」に従い、データ連携ツールを用いてCSV/TSV形式のファイルに変換して扱う
    - 技術的には、Frameworkに独自実装して、バイナリデータを入力させることは可能だが、実装や品質確保のコストがかかる
  - マルチレイアウト/マルチレコードフォーマットの場合は注意が必要
    - 「入出力データの判断ポイント」に従った形に変換が必要

参考情報

参考情報

OSS Consortium

- 開発元リンク
  - Asakusa Frameworkコミュニティサイト
    http://www.asakusafw.com/
  - Asakusa Framework ドキュメント
    http://asakusafw.s3.amazonaws.com/documents/latest/release/ja/html/index.html
  - Asakusa Framework ダウンロード
    http://www.asakusafw.com/techinfo/download.html
  - Asakusa Framework ソースリポジトリ
    https://github.com/asakusafw/asakusafw
- 開発者向け情報
  - Asakusa Frameworkメモ (Hishidama's Asakusa Framework Memo)
    http://www.ne.jp/asahi/hishidama/home/tech/asakusafw/index.html
- 勉強会情報
  - Asakusa Framework 勉強会
    http://asakusafw.connpass.com/