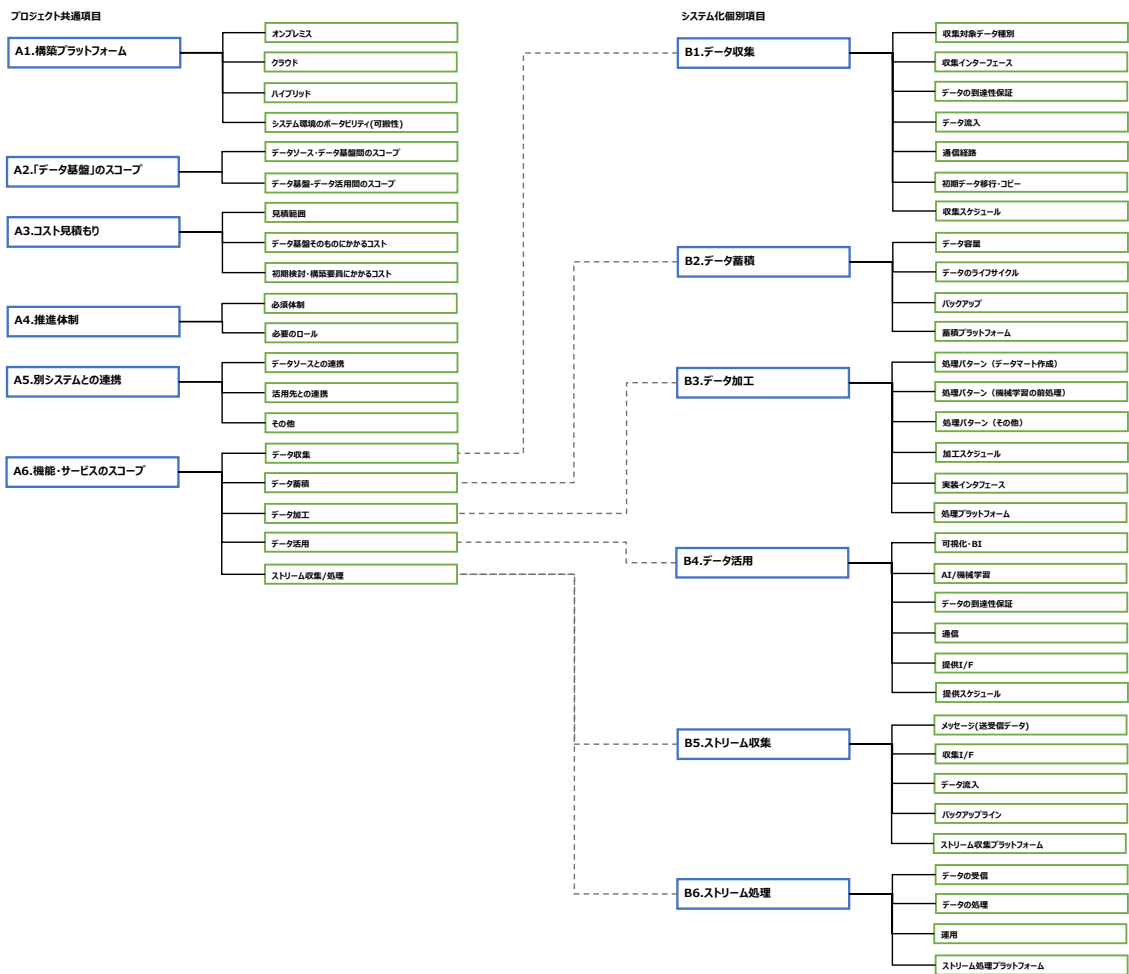


観点一覧

検討項目カテゴリの樹形図(中カテゴリまでのサマリ)



システム化個別項目

項目	大カテゴリ	中カテゴリ	中カテゴリの説明	小カテゴリ	小カテゴリの説明
B.1.1.1	データ収集	収集対象データ種別	データベースからデータ基盤内に取り入れるデータの種別	構造データ(Structured Data)	RDBMSのテーブルやCSVなど、テーブル構造で表すことが可能なデータ 例： ・CSV(Comma Separated Values) ・TSV(Tab Separated Values) ・システムログ
B.1.1.2				半構造データ(Semi-Structured Data)	XMLやJSONなど、内部で多次元の構造を保持可能なデータ。厳密な定義はないが、ここでは以下の性質をもつものとする ・(不可能ではないが)SQLでの扱いが難しい ・後方互換性を保ちながらスキーマの変更や、前後スキーマの混在が可能 ・情報量を失わずにテキストで表すことが可能
B.1.1.3				非構造データ(Unstructured Data)	テキストファイル(*.txt)、画像(JPG、PNG、BMP、GIF)や音声(WAV、MP3、AAC)、動画(AVI、MP4、etc...)など センサーなどのIoTデバイスにみられる。独自フォーマットによりシリアライズされたフォーマット、シリアライズ/デシリアライズのためにビットアイン表など別途フォーマットが必要
B.1.1.4				特定のアプリケーション向けフォーマット	Word(*.docx)やExcel(*.xlsx)、PDFなど用途が限定的なアプリケーションで扱った形式
B.1.2.1		収集インターフェース	データベース毎に、どのようなインターフェース経由でデータを収集するか	バッチ連携	対向システムで一定期間データを蓄積し、データ基盤側に送信する 例：日次(夜間)バッチによる連携、時次バッチでの連携
B.1.2.2				ストリーム連携	システムロギング、RDBMSのトランザクションログ、センサーデータの発生など、対向システムでのデータ発生と同時にデータ基盤へデータが送信される連携方式
B.1.2.3				PUSH型/ PULL型	■PUSH型 データベースがデータを送信 例：データ基盤はデータを受信するための口を待ち、データベースのタイミングでデータを送信する。 ■PULL型 データ基盤がデータベースにアクセスしデータを取得 例：ETLツールがRDBMSにアクセスしデータ基盤にデータを格納する
B.1.2.4				対向システムへの負荷	収集時の対向システムへの負荷(CPU負荷、I/O負荷、通信量)
B.1.2.5				プロトコル	■ミドルウェアに依存しない汎用的なプロトコル例 ・FTP、SFTP(ファイル転送) ・HTTP/HTTPS(主にRestAPI) ・MQTT、MQTT(SIoTなどデバイス向け) ■ミドルウェア依存のプロトコル例 ・JDBC/ODBC ・FluentdやKafkaなどミドルウェアによる転送
B.1.3.1		データの到達性保証	連携するデータ毎にデータ連携時の欠損・重複についての対応・責任分界点	データ到達の責任分解	データ送信の責任をデータベースとデータ基盤どちらが持つか
B.1.4.1		データ流入	利用するデータの流入量	データ流入量	データベースから収集する収集対象データ量(OTByte/day、OByte/hourなどで表される)
B.1.4.2				データ流入量のゆらぎ	データ流入量が時間軸で変動するか
B.1.5.1		通信経路	データ収集のための外部システムとの接続経路	帯域	データベースのシステムへデータ基盤間のネットワーク帯域
B.1.5.2				暗号化	通信経路の暗号化
B.1.6.1		初期データ移行・コピー	既存システムがある場合、システムからデータ基盤へのデータ移行についての検討項目	移行データ量	既存システムから移動させるデータの総量
B.1.6.2			収集元がデータ基盤と平行で継続運用される業務システムの場合は、移行ではなくコピーという表現もできる	移行スケジュール	既存システムからの移行計画
B.1.6.3				移行方式	既存システムからのデータ移行方式
B.1.7.1		収集スケジュール	データ収集のスケジュール	ジョブ管理システム連携	データ収集を起動する機能との連携
B.1.7.2				業務SLA確認	データ収集の開始終了時間など業務的な要件
B.2.1.1		データ蓄積	データ蓄積	データ容量	データベースから収集するデータの容量
B.2.1.2				活用データの容量	データを加工し、活用に受け渡す状態のデータの容量
B.2.1.3				中間データの容量	元データから活用データに加工する間で、アプリケーションで必要になる中間状態のデータの容量
B.2.1.4				データの圧縮	元データ、活用データ、中間データをそれぞれ圧縮するか
B.2.2.1		データのライフサイクル	データ毎にどの期間データを保存するか、削除するか	データ保管期間	元データ、活用データ、中間データをそれぞれの保管期間
B.2.2.2				アーカイブの有無	低コストかつI/O性能の低いストレージにデータを連携するか
B.2.2.3				アーカイブ保管期間	アーカイブデータの保管期間
B.2.3.1		バックアップ	システム障害等に対するデータ消失を防ぐ	バックアップの有無	データのバックアップを取得するか否か
B.2.3.2				チェックポイント	特定の時点のストレージの状態をチェックポイントとして記録し、切り戻すことを可能とする機能 アプリケーションエラー、オペレーションミス等によるデータ消失からデータを復旧させることが可能になる場合がある
B.2.3.3				災害復旧(DR)の有無	災害対策用サイトを有するか否か
B.2.3.4				DRサイトへの転送方式、保存期間、ロケーション	本番環境からDRサイトへのデータ転送方式、データの保管期間、DRサイトを構築する地理的な場所
B.2.4.1		蓄積プラットフォーム	蓄積機能を実現するためのプラットフォーム	導入製品	データ蓄積基盤の中心として採用するストレージ製品
B.3.1.1	データ加工	処理パターン(データマート作成)	データ基盤上で中間処理を行う処理の代表的パターン	フルタング	条件に合致データを抽出
B.3.1.2				集計・一時集計処理	時間帯やグループ別の集計処理を行う。
B.3.1.3				名寄せ	複数のデータから、同一人物や同一企業のデータを1つにまとめる
B.3.1.4				マスターデータ結合	複数のマスターデータの結合
B.3.1.5				順序制御	項目に応じたデータの並べ替え
B.3.1.6				重複排除	重複データを取り除く
B.3.2.1		処理パターン(機械学習の前処理)	機械学習等の分析アルゴリズムで処理する前に必要となる前処理の代表的パターン	(機械学習アルゴリズムのインプットとするための)前処理	クレンジング、正規化、標準化、ベクトル化など
B.3.2.2				テキスト(自然言語)処理	特徴量抽出
B.3.2.3				非構造データ処理	画像や音声、動画等のデータからの特徴量抽出
B.3.3.1		処理パターン(その他)	上記パターンに含まれないデータ加工処理パターン	検索インデックス作成	検索インデックスの作成
B.3.4.1		加工スケジュール	バッチ処理のワークロードでの加工処理が必要な場合、どのようにスケジュールを割り当てるか	-	-
B.3.5.1		実装インターフェース	データ加工処理を実装するためのアプリケーション実装インターフェース	SQL	SQLまたはSQLライクな実装言語による開発
B.3.5.2				プログラム	SQL以外の実装言語による実装。ノートブックなどUI経由で実装したものを含む。
B.3.5.3				ETLツール	ETLツールに内包されたデータ加工機能
B.3.6.1		処理プラットフォーム	加工機能を実現するためのプラットフォーム	加工で実施する内容	クレンジング、前処理、結合処理、フルタング処理、DM作成処理など、処理パターンで定義した内容。
B.3.6.2				導入製品	加工基盤に導入する製品(ETL、DWH、Hadoop,...) 実施したい処理に応じて、複数導入することもある
B.3.6.3				製品の使い分け	複数の製品を導入する場合の使い分け指針

システム化個別項目

項目	大カテゴリ	中カテゴリ	中カテゴリの説明	小カテゴリ	小カテゴリの説明		
B.4.1.1	データ活用	可視化・BI	データを分析し、経営上の意思決定に役立てる	BI機能の有無	BI機能を利用するか		
B.4.1.2				BIツールの種類	エンタープライズBI/セルフBIといった分類が存在するか		
B.4.1.3				BIデータアクセス経路	アクセス対象のデータが格納されるDBはデータ基盤の内か外か		
B.4.1.4				BIアクセス制御	アクセス元のユーザのアクセス制御		
B.4.2.1		AI/機械学習	データから学習モデルを生成し、推論(予測)を行う	データから学習モデルを生成し、推論(予測)を行う	AI/機械学習機能の有無	AI/機械学習機能を利用するか	
B.4.2.2					学習環境	学習モデルを生成する環境	
B.4.2.3					推論環境	学習モデルをデプロイ/予測を実行する環境	
B.4.2.4					機械学習提供ツール/プラットフォーム	どのようなツール/プラットフォームをベースとしてライブラリ/機能開発を実施するか	
B.4.3.1		データの到達性保証	活用先毎のデータ提供時の欠損・重複についての対応	活用先毎のデータ提供時の欠損・重複についての対応	データ欠損・重複時の対応	活用先へ提供するデータに問題があったときどう対応するか	
B.4.3.2					対向システムへの負荷	活用先へ提供時の対向システムへの負荷(CPU負荷、I/O負荷、通信量)	
B.4.3.3					データ流出量	活用先への提供データ量	
B.4.4.1			通信	活用先へデータ提供時のネットワーク	活用先へデータ提供時のネットワーク	帯域	確保すべきネットワーク帯域
B.4.4.2						暗号化	暗号化要件があるか
B.4.5.1			提供I/F	活用先毎に、どのようなインターフェース経由でデータを提供するか	活用先毎に、どのようなインターフェース経由でデータを提供するか	-	-
B.4.6.1	提供スケジュール	活用先毎のデータ提供スケジュール		-	-		
B.5.1.1	ストリーム収集	メッセージ(送受信データ)	ストリームデータ(断続的・連続的に生成されるデータ)	メッセージサイズ	1メッセージ当たりの容量		
B.5.1.2				データ形式	メッセージのファイルフォーマット ファイルの中身の構造		
B.5.1.3				圧縮有無	メッセージの圧縮の有無		
B.5.1.4				メッセージ種別	収集するメッセージの種類		
B.5.1.5				メッセージの順序保証	所定のメッセージ順序を守って処理するかどうか		
B.5.1.6				メッセージのカーディナリティ	データの取り得る値のばらつきと偏り		
B.5.1.7				メッセージのスキーマの変化	スキーマの変更(カラム増減、型の変更など)があるかどうか		
B.5.1.8				メッセージの保存	メッセージ内部のメッセージの保存		
B.5.1.9				収集時のメッセージ加工	収集するタイミングでメッセージを加工するか		
B.5.2.1		収集I/F	データソース毎に、どのようなI/F経由でデータを収集するか	データソース毎に、どのようなI/F経由でデータを収集するか	プロトコル	ストリーム収集で利用するプロトコル	
B.5.2.2					送信クライアント数	データソース側のメッセージ送信を行うクライアント数	
B.5.2.3					基盤・フレームワーク	データソース側のメッセージ送信を行うAPIの基盤・フレームワーク	
B.5.2.4					リアルタイム性の担保	高スループット、低レイテンシなどの程度実現するか	
B.5.2.5					メッセージ送達保証のセマンティクス	送達保証のレベルとメッセージロスとの責任分界	
B.5.2.6					リアルタイム性とメッセージ送達保証のバランス	リアルタイム性(高スループット、低レイテンシ)とメッセージ送達保証はトレードオフであることがあるため、そのバランスを考慮する	
B.5.2.7					タイムアウト設計	送信元のタイムアウト設計と受信側のタイムアウト設計	
B.5.3.1		データ流入	上流システムから受信するデータの流入条件	上流システムから受信するデータの流入条件	スループット	単位時間あたりに送信するメッセージ数	
B.5.3.2					データ流入のワークロード	メッセージ送信の間隔	
B.5.3.3					データ流入量の考え方	ワークロードに応じて、データソース側とスループットの考え方があるか	
B.5.3.4					拡張性	データ流入量にあわせた拡張性	
B.5.3.5					バーストラフィックの有無	バーストラフィックが発生する可能性があるか	
B.5.3.6					メッセージ遅延	メッセージが遅れて送信・受信されることがあるか	
B.5.4.1		バックアップライン	ストリーム収集の副系統	ストリーム収集の副系統	-	-	
B.5.5.1	ストリーム収集プラットフォーム		ストリーム収集機能を実現するためのプラットフォーム	導入製品	ストリーム収集基盤として採用する製品		
B.6.1.1	ストリーム処理	データの受信	メッセージバッチからのデータ受信	スループット	単位時間あたりに受信するメッセージ数、量		
B.6.1.2				メッセージ受信のワークロード	メッセージバッチからのメッセージ受信のワークロード		
B.6.1.3				大量メッセージ流入	想定以上のメッセージ流入への対応		
B.6.1.4				チェックポイント	途中経過を記録するための一時的なデータ保存をするかどうか		
B.6.2.1		データの処理	受信したデータの処理	受信したデータの処理	ステート管理	処理の状態管理	
B.6.2.2					バックプレッシャー	バーストに対する処理抑制なし抑制の伝播	
B.6.2.3					アプリケーションの更新	継続的に動作し続けるアプリケーションのアップデート	
B.6.2.4					順序保証	所定の順序を守り処理をするかどうか	
B.6.2.5					メッセージ処理保証のセマンティクス	ストリーム処理では故障発生時などに仕掛中の処理が途絶えることがある。そういった際どのように処理継続するかを決める必要がある。(例：最大でも1回処理する、最低1回処理する、ただ一度処理する)	
B.6.2.6					性能	ストリーム処理の性能(例：スループット、レイテンシ)	
B.6.2.7				処理の並列度	スループットを向上するため処理を並列化することがあるかその度合い		
B.6.2.8				データ流入量	データ流入量にあわせた拡張		
B.6.2.9			データ永続化の有無	ストリーム処理対象のデータ、中間データ、処理した結果を永続化するかどうか			
B.6.2.10			処理ロジック	ストリーム処理のロジック			
B.6.3.1	運用	ストリーム処理基盤の運用	ストリーム処理基盤の運用	業務閉塞の有無	メンテナンス等で業務閉塞するか		
B.6.4.1		ストリーム処理プラットフォーム	ストリーム処理を実現するためのプラットフォーム	導入製品	ストリーム処理基盤として採用する製品		