

クラウド時代のファイル転送の価値

株式会社セゾン情報システムズ

セールスエンジニアリング部

福永 健一

ファイル転送って、どんなイメージをもっていますか？

単純・原始的

単なるコピーじゃないの？

FTP

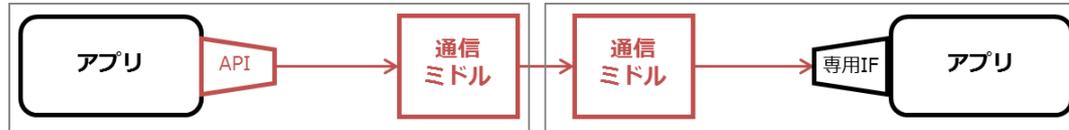
古くさい仕組み

わざわざお金をかけるの？

ダウンロードのこと？

システム間の連携方式って、何があるのか？

プログラム連携



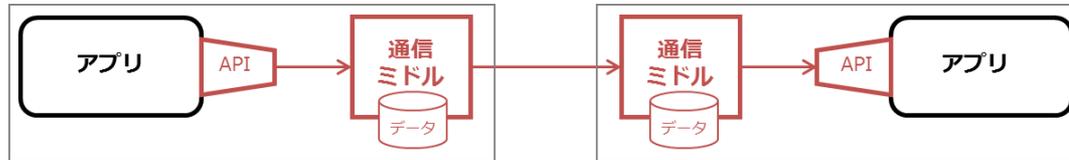
RPC、Corba、DCOM、
独自プロトコル (Socket)

サービス連携



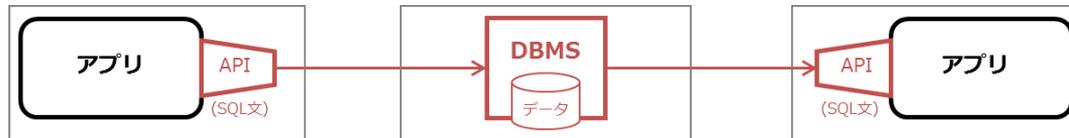
SOAP、REST、ESB、
API Gateway

メッセージ連携



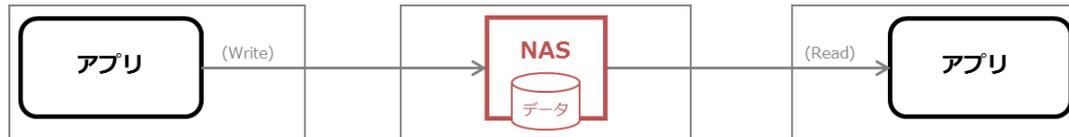
JMS、MQ、MQTT

DB連携



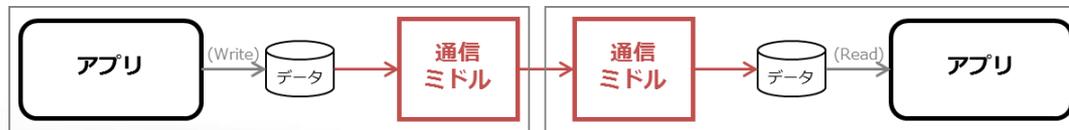
RDBMS

ストレージ連携



NAS

ファイル連携



FTP、SSH、全銀手順、
HULFT

連携方式の特徴①

バックグラウンド 一括処理、定期処理に適しています。

連携型： ⇒ 時間内の完了が重要、**コスト低**

リアル連携型： 対話型の処理に適しています。

⇒ レスponse時間が重要、**コスト高**

バックグラウンド連携型

メッセージ連携

DB連携

ストレージ連携

ファイル連携

リアル連携型

プログラム連携

サービス連携

連携方式の特徴②

ジョブ連携型：アプリはRead/Writeのみで連携は考慮不要です。
連携処理は別途ミドルウェア **設定** だけで実装できます。

API連携型：アプリ自身で連携処理を **作り込む** 必要があります。

ジョブ連携型

DB連携

ストレージ連携

ファイル連携

API連携型

プログラム連携

サービス連携

メッセージ連携

連携方式の特徴③

疎結合：システムを変更しても連携先のシステムが影響を受けにくい。
⇒ビジネスの変化に素早く対応

密結合：システムを変更すると連携先システムも同時に変更が必要。
⇒変更容易性よりもシステム性能を優先

疎結合

サービス連携

メッセージ連携

DB連携

ストレージ連携

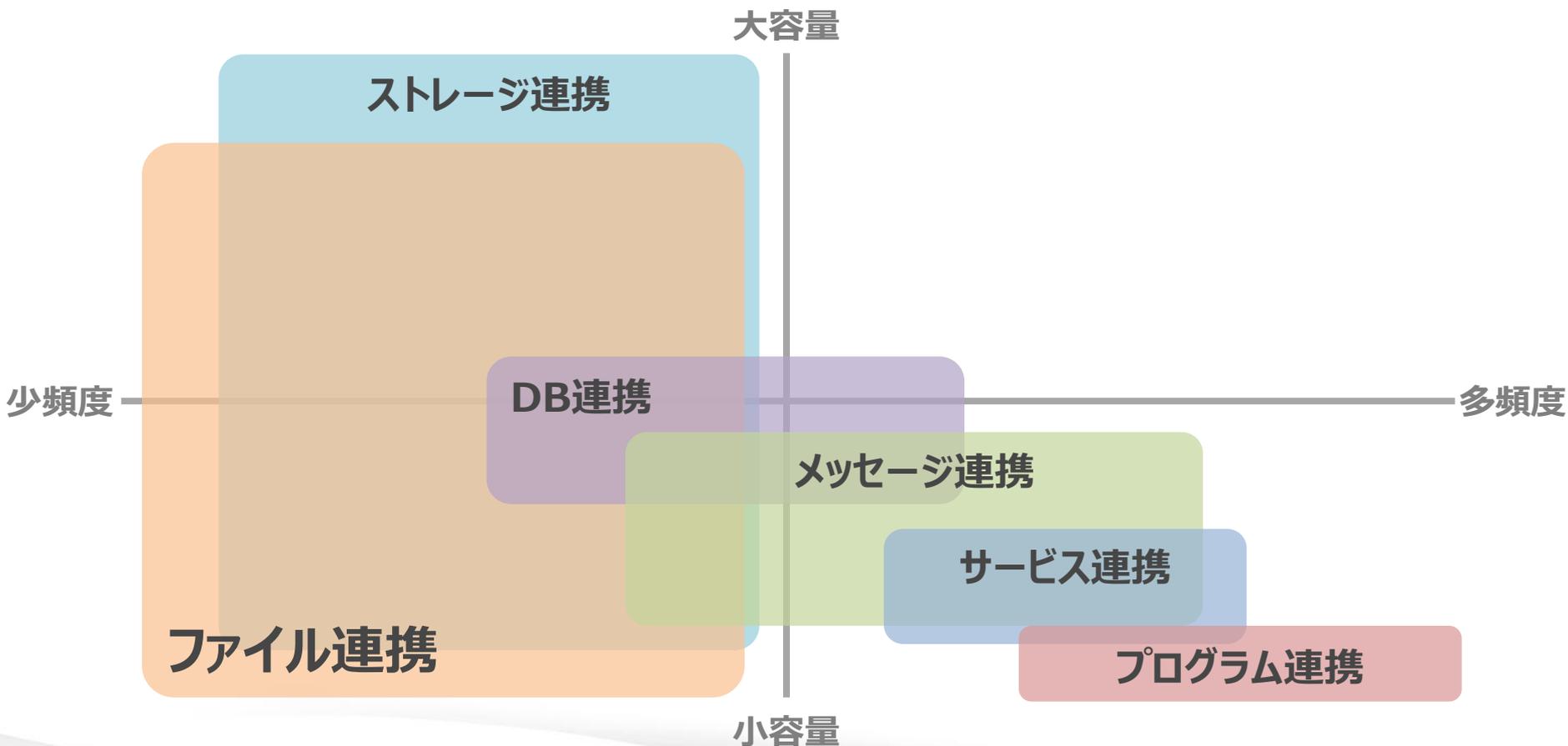
ファイル連携

密結合

プログラム連携

連携方式の特徴④

連携する頻度とデータ量によって、各連携方式に向き・不向きがあります。
適切な方式を選択することによって、適切なコストで連携できます。



ファイル転送の使いどころ

■ 大容量データの一括連携

ビッグデータの連携はファイル転送の高速連携が必須

100MB～GBサイズの連携は、アドバンテージが大きい

■ バックグラウンド型の連携

分単位のデータ反映で十分な場合、簡単・低コストで実現可能

■ 最もシンプルな疎結合型の連携

シンプル⇒低コストでの導入が容易

少額投資で連携できるのでスモールスタートに最適

ファイル転送の課題

システム間連携でファイル転送を採用した場合・・・
単純にファイルを送るだけで十分でしょうか？

情報漏えいは
大丈夫かな？

データが壊れてい
ないよね？

様々なプラットフォームと
つながられるのかな？

そもそも簡単に
システム連携できるの？



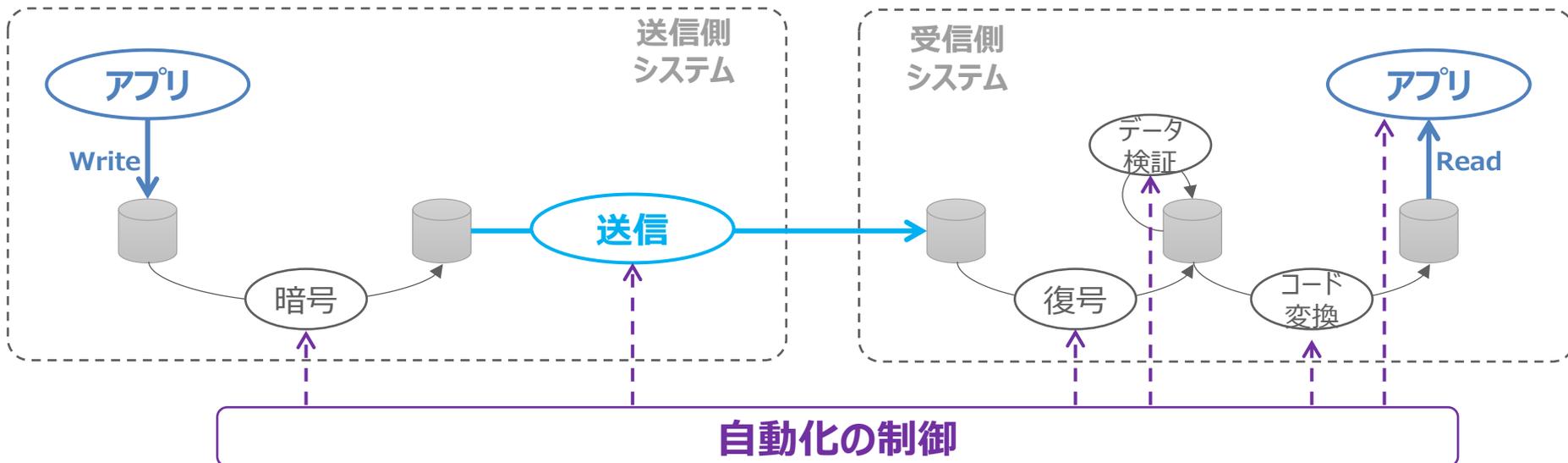
転送機能だけだと困りませんか？

たとえば、FTPを使った場合・・・

安全性を担保するために暗号処理を組み込み
データ保障のために検証処理を組み込み

.....

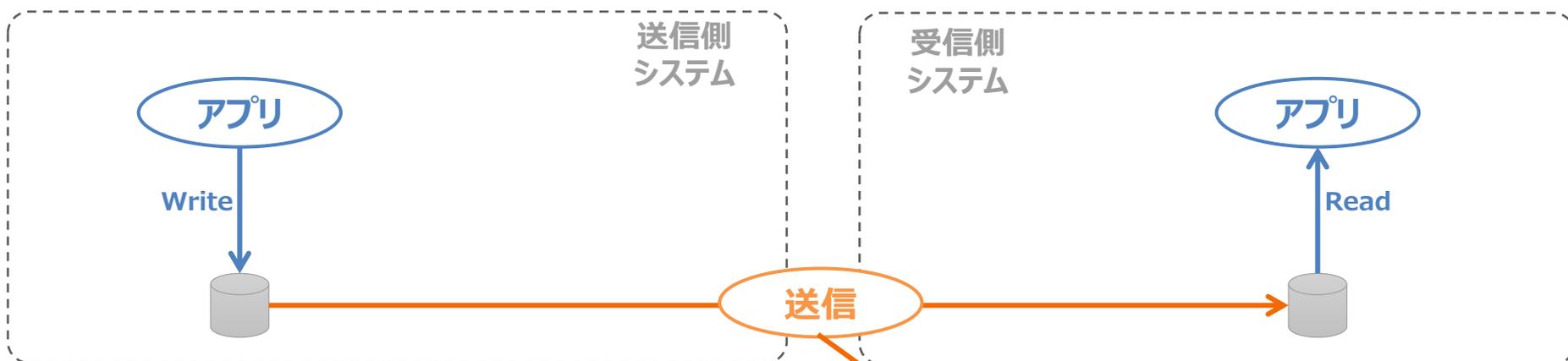
こんなシステム作っていたりしませんか？



ファイル転送の理想像

もっとシンプルに連携できませんか？

やりたいことは確実なシステム間の連携だけ
わざわざ作り込みたくない、**設定だけ**で実現したい



全部乗せ

暗号化 (データ漏えい)
データ保障 (欠落・改ざん)
コード変換 (SJIS \leftrightarrow UTF8)
アプリ間の自動連携

【参考例】

前提条件

処理概要：Windows-Linux間のファイル転送(10件/1日)
 ファイル受信後に業務処理を実行する。
 障害時の復旧作業：1.0人日（障害件数：4回/1年）

- ・ノンプログラミング
- ・テスト作業の短縮

比較項目(システム構築)	FTP	ツール使用	備考
構築 (インストール)	0.0	0.3	インストーラによるインストール(ツール)
(環境設定)	5.0	2.7	サーバーおよび各種ネットワーク設定(ツール・FTP)
設計	12.0	6.0	・転送プログラムの設計(FTP) ・バッチ処理の設計(ツール・FTP)
開発 (転送確認、存在チェック)	4.0	0.0	バイト数チェック、ファイル存在チェックの機能実装(FTP)
(文字コード変換)	10.0	0.0	文字コード変換機能の実装(FTP)
(転送履歴管理)	8.0	0.0	ファイル転送履歴管理機能の実装(FTP)
(アプリケーション自動連携)	6.0	0.0	転送ファイルの確認→業務処理の自動実行機能(FTP)
検証 (システムテスト)	10.0	8.0	・手組みプログラムのテスト(FTP) ・システムテスト(ツール・FTP)
合計	55.0	17.0	人日

FTPの構築工数の約70%減！

比較項目(保守・運用)	FTP	ツール使用	備考
運用 (オペレーション)	8.5	4.0	転送数×障害発生率×再送までに費やす人的コスト(FTP) 障害発生率1.25倍、再送コスト1.5倍(HULFTとの比較)
合計	8.5	4.0	人日

FTPの保守・運用工数の約50%減！

※HULFTは上記のほか別途ライセンス費用および保守費用がかかります。

ファイル転送が備えるべき12の要件

安全

要件1 機密性の保障

要件2 不正アクセスの防止

要件3 トレーサビリティの保障

要件4 サービスの堅牢性

確実

要件5 送達の保障

要件6 データの保障

要件7 自律的なリカバリー

柔軟

要件8 疎結合

要件9 システム間の差異を吸収

要件10 ネットワーク帯域の制御

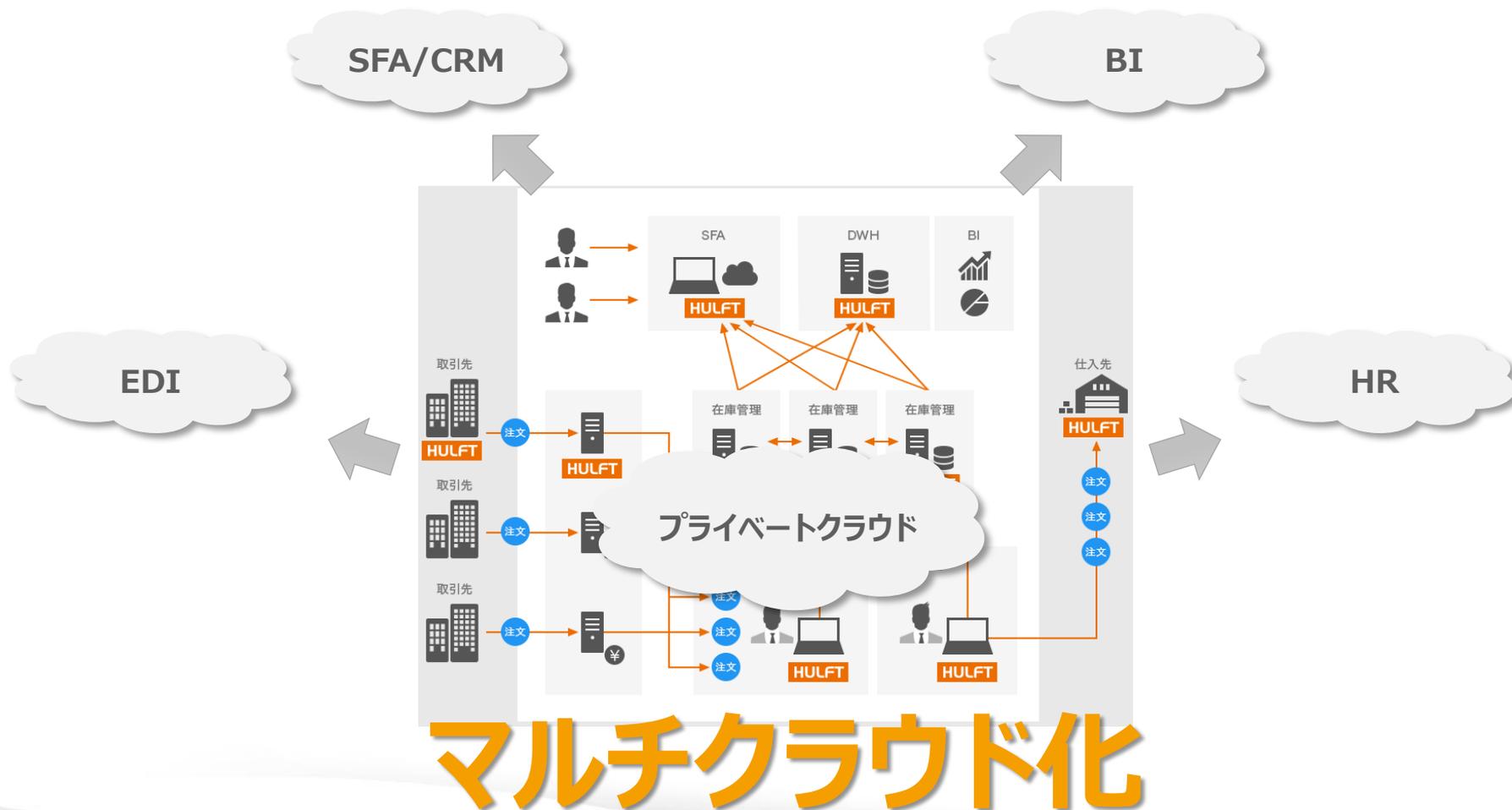
簡単

要件11 運用の容易性

要件12 連携の容易性

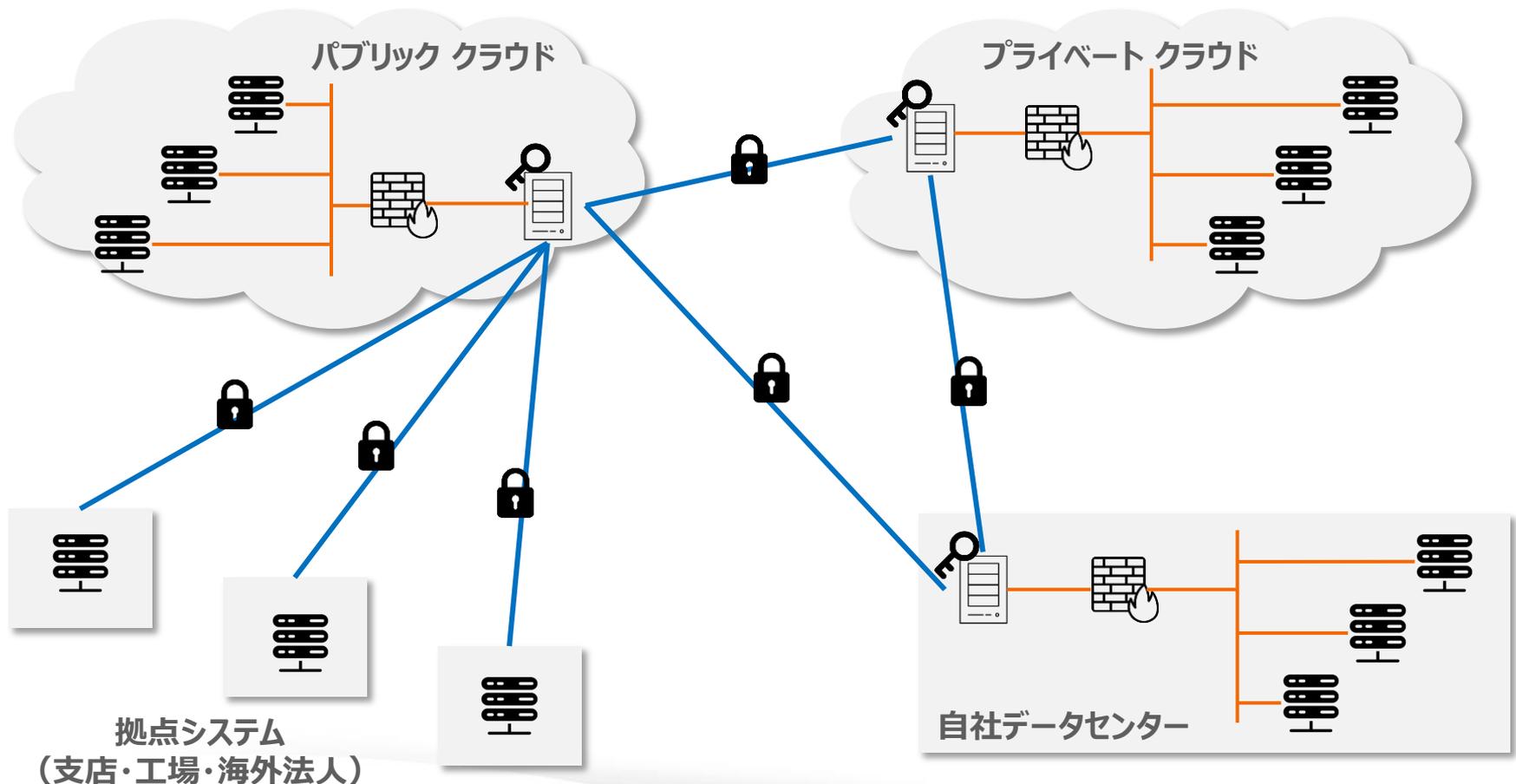
クラウド時代のシステム構成

クラウド時代のシステム構成はどのようになるのでしょうか？



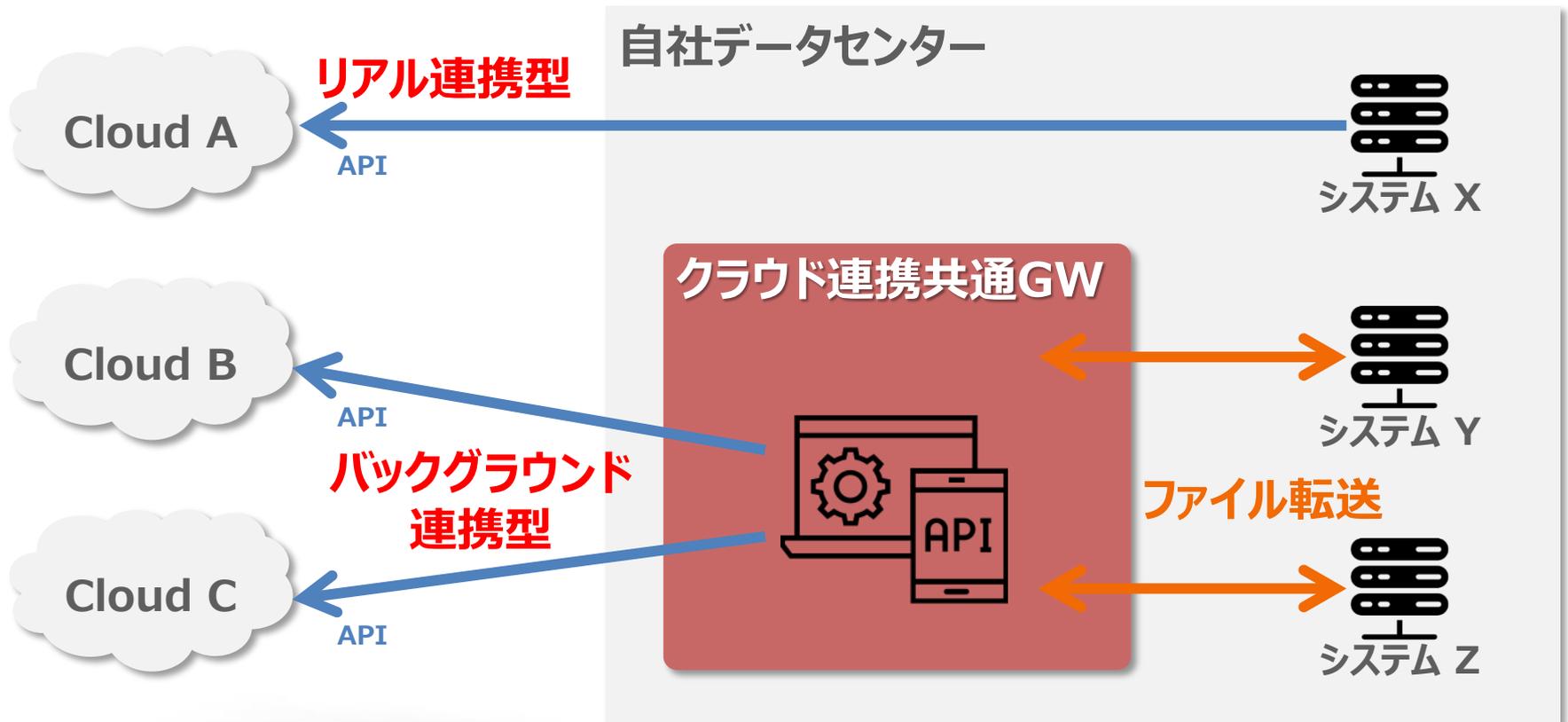
クラウド時代はデータ処理が分散化します

クラウド間、拠点間のデータ連携がますます増加します。
マルチクラウド環境での最適な連携方法は？



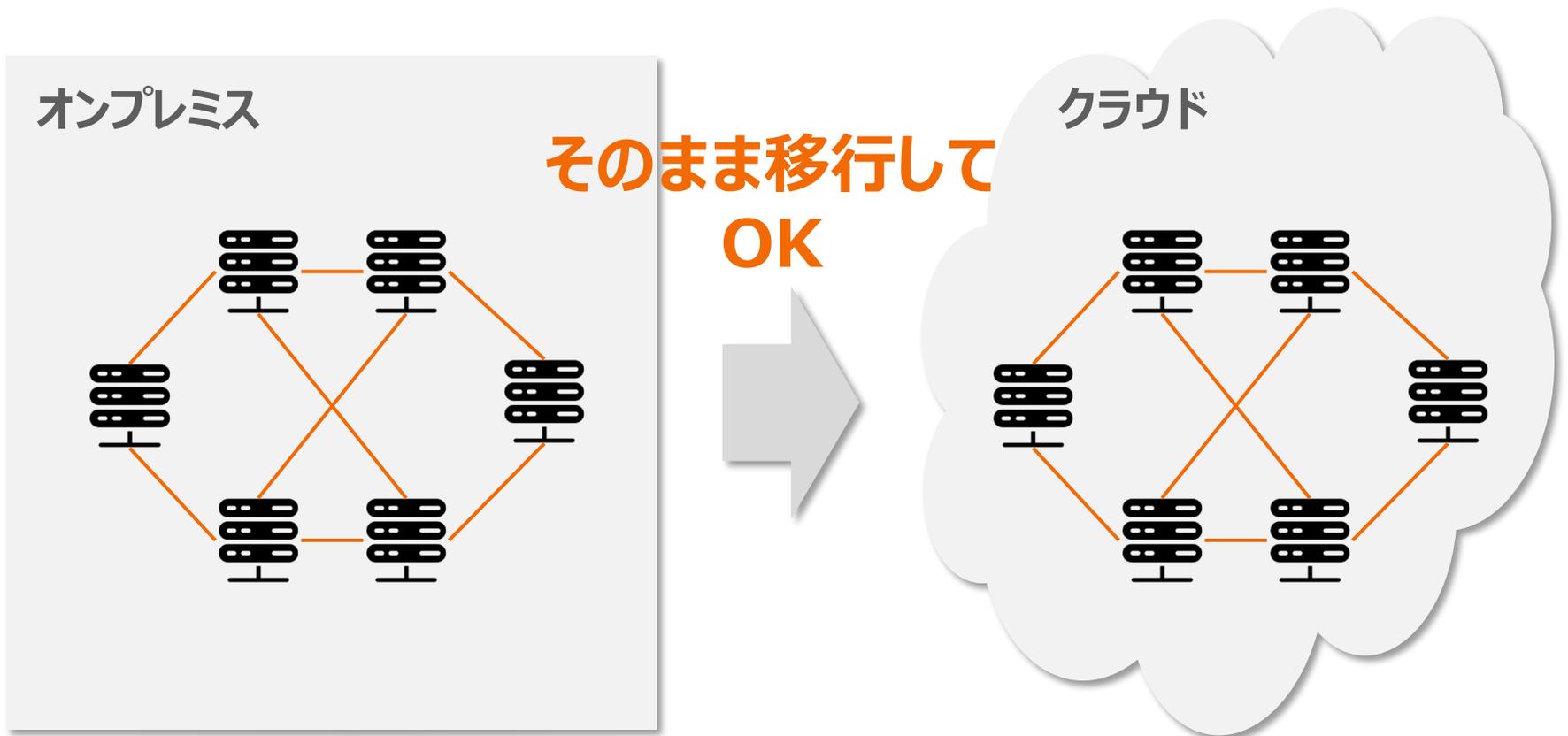
パブリッククラウドとの経済的な連携方法は？

パブリッククラウドとの連携にはAPI対応が必要になります。
連携するたびに作り込みますか？



クラウドに移行するとファイル転送は不要？

ファイル転送の価値は、クラウド時代も変わりません。
使いどころを押さえておくのが賢い選択です。



クラウド時代のファイル転送の価値とは

■ ファイル転送のアドバンテージは

ビッグデータの高速転送

バックグラウンド処理を低コストで実現

■ ファイル転送のクリアすべき4つの課題

安全、確実、柔軟、簡単

■ クラウド時代のファイル転送

マルチクラウド環境のセキュアなつながり

パブリッククラウドと容易につながる

低コストで素早くつながること