

# 大量TIFFデータの公開を可能に！

## HG/DDS(ダイナミック・ドキュメントサーバ)のご紹介



2009年11月27日

株式会社ハイパーギア

## TIFF画像公開の問題点

**1** 簡単に修正できてしまう

**2** セキュリティがない

**3** Windows OSの違いにより、ビューワが不統一になり、問題が発生

# PDFで問題を解決



1 どんなPCでも閲覧が可能

2 ファイル単位でセキュリティ設定  
編集の可否、印刷の可否が可能

3 世界標準の文書フォーマット

# 大量電子データの問題点は



◆ 全てのデータをAcrobatでPDFに変換すると

**1** 速度が遅い

**2** データが2倍(元TIFFとPDF)

**3** システムが自由に組めない

## PDF変換JavaBeans™サーブレット

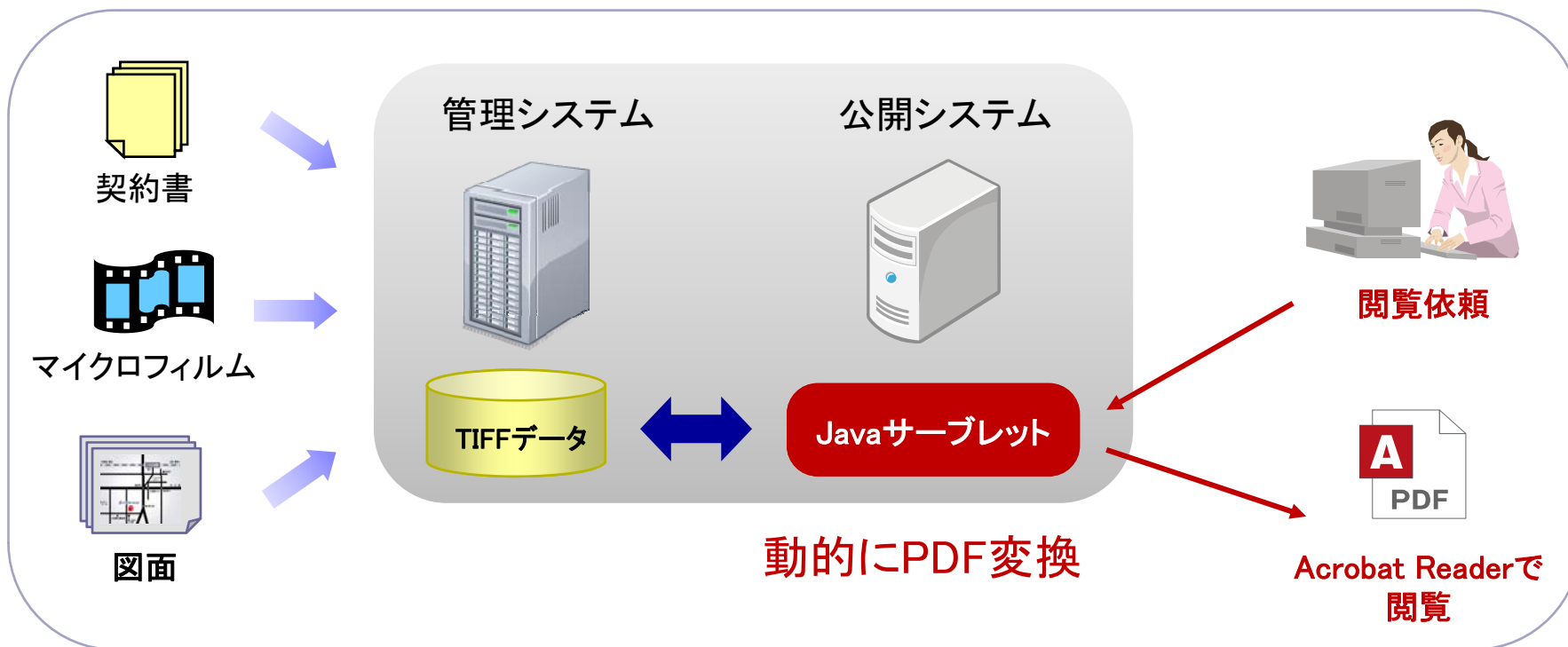
**1** 閲覧時に動的にPDF変換

**2** ファイル単位でセキュリティ設定  
編集の可否、印刷の可否が可能

**3** システムに組み込み可能

# HG/DDS概要図

## ◆ HG/DDS (PDF変換JavaBeans™サーブレット)



閲覧希望時に動的にPDF変換を行い  
データの2重化問題をクリア

## 閲覧時に動的にPDF変換

閲覧リクエストが  
あった時の、  
リアルタイム処理

高速  
1ページあたり、  
28ミリ秒(A4,200dpi)

多彩なフォーマット  
に対応  
モノクロ/カラー対応  
TIFF(JPEG)  
MC-PDF対応  
2色 高圧縮 PDF

公開のためのデータを用意する必要がないので、ディスク容量が倍にならない。  
元TIFFデータの変更時でも、公開用データを準備する必要がない。

MC-PDF,高圧縮PDFは、上記変換時間以上必要です。

# ファイル単位でセキュリティ設定

## 標準セキュリティ

40bit/128bit RC-4

パスワード

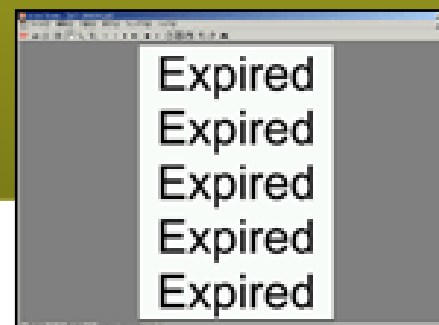
印刷禁止

変更禁止

## 独自セキュリティ

ダウンロード禁止

きめ細かなセキュリティが、動的に設定できるので、情報の公開が安全に出来る。



## システムに組み込める

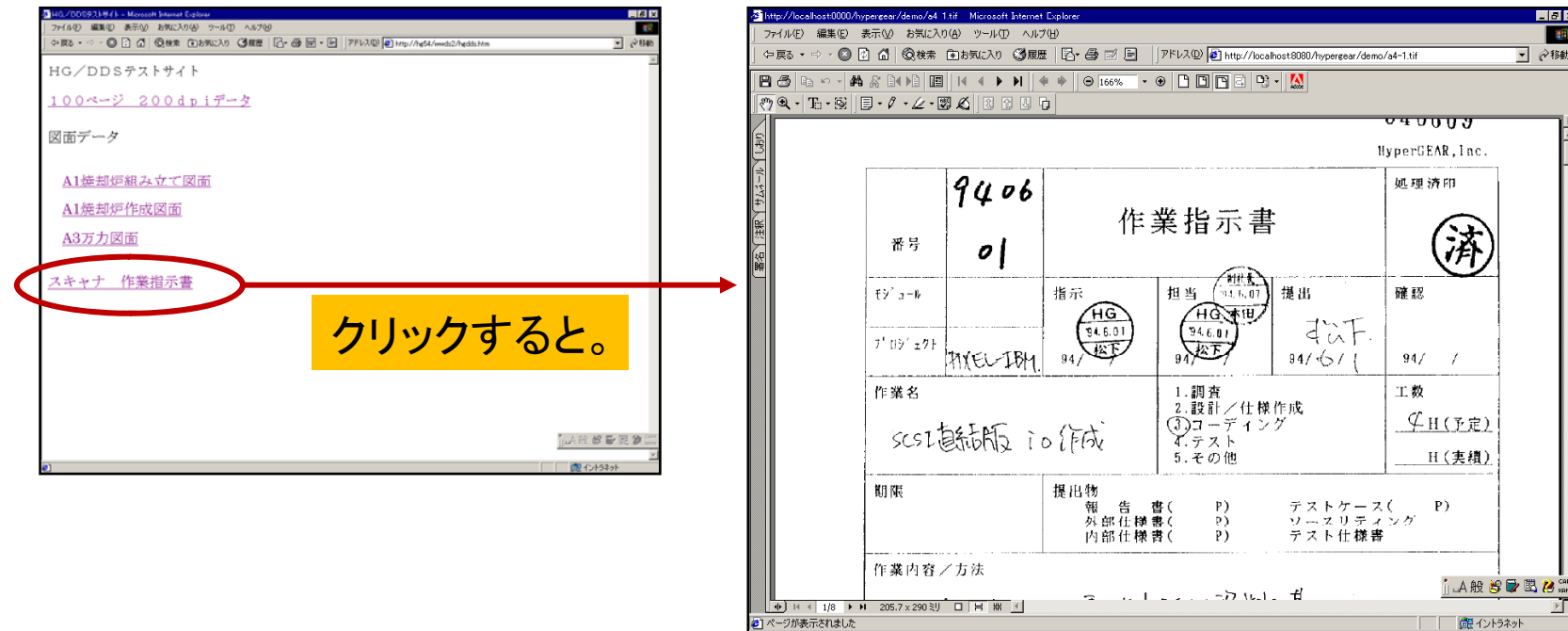
100%Java™により開発された、java Beans™コンポーネントサンプルサーブレット付き

プラットフォームに依存しない。  
小規模なPCベースシステムから、大規模UNIXサーバまでスケーラブルにシステム構築

目的に応じた高性能なサーブレットを容易に開発

セキュリティ設定等は、Windows版のみでの対応です。  
LinuxやUnixでは、PDF変換は可能ですが、セキュリティ設定は未対応です。

# サンプル画面



The image shows two browser windows. The left window displays a menu with the following items:

- HG/DDSテストサイト
- 100ページ 200dpiデータ
- 図面データ
  - A1接卸伊組み立て図面
  - A1接卸伊作成図面
  - A3万力図面
  - スキャナ 作業指示書

A red circle highlights the 'スキャナ 作業指示書' item, with a red arrow pointing to a yellow box containing the text 'クリックすると。' (When clicked).

The right window shows a detailed work instruction form titled '作業指示書' (Work Instruction) for '940601'. The form includes the following fields:

- 番号: 940601
- 処理済印: (済)
- 指示: HG 94.6.01 松下
- 担当: HG 94.6.01 松下
- 提出: 松下 94/6/1
- 確認: 94/ /
- 作業名: SCSI接続板作成
- 工数: H(予定)
- 提出物: 報告書(P), 外部仕様書(P), 内部仕様書(P), テストケース(P), リソースリテイング, テスト仕様書

URLは、<http://hg54/hypermear/demo/a4-1.tif>

TIFFを表示しようとする、Acrobat Readerが起動し、PDFで表示される。

# コマンド例

メソッド	全29
変換	13
セキュリティ	12
画像	4

クラス TiffToPdfBean2 public class  
TiffToPdfBean2  
extends java.lang.Object

コンストラクタの概要  
TiffToPdfBean2()

## メソッドの詳細

convert

public void convert() throws java.lang.Exception  
変換を実行し変換結果を返却する。

setSrcFileName

public void setSrcFileName(java.lang.String  
fname)  
変換元ファイル名の設定。

setDocTitle

public void setDocTitle(java.lang.String title)  
文書タイトル名の設定(PDF文書情報)。

setSecLevel

public void setSecLevel (int secLevel)  
PDF暗号化レベルの指定。  
secLevel – 0 , 40 128 の何れかを指定する。

setSecURL

public void setSecURL (java.lang.String url)  
Webダウンロード後の閲覧禁止セキュリティを設定する。指定したURL上から直接閲覧する事は可能だが、ローカルディスク上にダウンロード後に表示すると全体が「Expired」で覆われ、閲覧することができない。

パラメータ:

url – Web掲示するURLパスを指定する

例: http://www.aaa.com/data/\*.pdf などのワイルドカード指定も可能

# サンプルサーブレット付き

SampleServlet.java (HTTP Get によるURLパラメータから動作させるサンプル)  
SampleServlet2.java (特に、メソッドの呼び出す手順を記述したサンプル)

## 作成手順

1. PDF変換Bean のインスタンスを生成する
2. 変換パラメータの設定を行う
3. 出力ストリームの設定を行う
4. 変換実行
5. エラー処理

```
SampleServlet2.java - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

public class SampleServlet2 extends HttpServlet {
    private static final String CONV_FILENAME = "C:\\SampleDir\\Data\\test.tif";
    private static final String CONV_WORK = "C:\\SampleDir\\ConvWork";
    private static final String LOG_FILENAME = "C:\\SampleDir\\conv.log";
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        ////////////////////////////////////////////////////////////////////
        // 1. PDF変換Bean のインスタンスを生成する
        ////////////////////////////////////////////////////////////////////
        TiffToPdfBean2 tpb = new TiffToPdfBean2();
        // 2. 変換パラメータの設定を行う
        ////////////////////////////////////////////////////////////////////
        // 2. 1. 必須項目の設定
        tpb.setSrcFileName(CONV_FILENAME); //変換元ファイル名の設定
        tpb.setWorkPath(CONV_WORK); //変換用ワークフォルダの設定
        // 2. 2. 任意設定項目の設定 (追加・省略可)
        tpb.setLogFileName(LOG_FILENAME); //変換ログファイルの設定
        tpb.setDocTitle("文書タイトル"); //文書情報: 文書タイトルの設定
        tpb.setSecLevel(TiffToPdfBean2.SEC_40BIT); //セキュリティ: 40ビット暗号化の設定
        tpb.setSecPrint(true); //セキュリティ: 印刷禁止設定
        ////////////////////////////////////////////////////////////////////
        // 3. 出力ストリームの設定を行う
        // Servlet の出力ストリームをPDF変換Bean の出力先に設定します
        ////////////////////////////////////////////////////////////////////
        res.setContentType("application/pdf");
        ServletOutputStream stm = res.getOutputStream();
        tpb.setDest(stm);
        ////////////////////////////////////////////////////////////////////
        // 4. 変換実行
        ////////////////////////////////////////////////////////////////////
        try {
            //変換実行
            tpb.convert();
        }
        ////////////////////////////////////////////////////////////////////
    }
}
```

社内TIFF管理している図面を、PDFにして公開へ

## 背景

### ●課題

- 1) 建造中のビル図面をビルオーナーに向けて公開するサービスを開始
- 2) 公開用のデータ等は持たずに、自社管理DBシステムと連携した公開システムをサーバレットで作成した。

### ●問題点

自社管理DBでは、TIFF形式が標準フォーマットになっているが、TIFF形式での公開では、表示できない。変更が容易にできてしまうなどの問題が発生。

## 提案

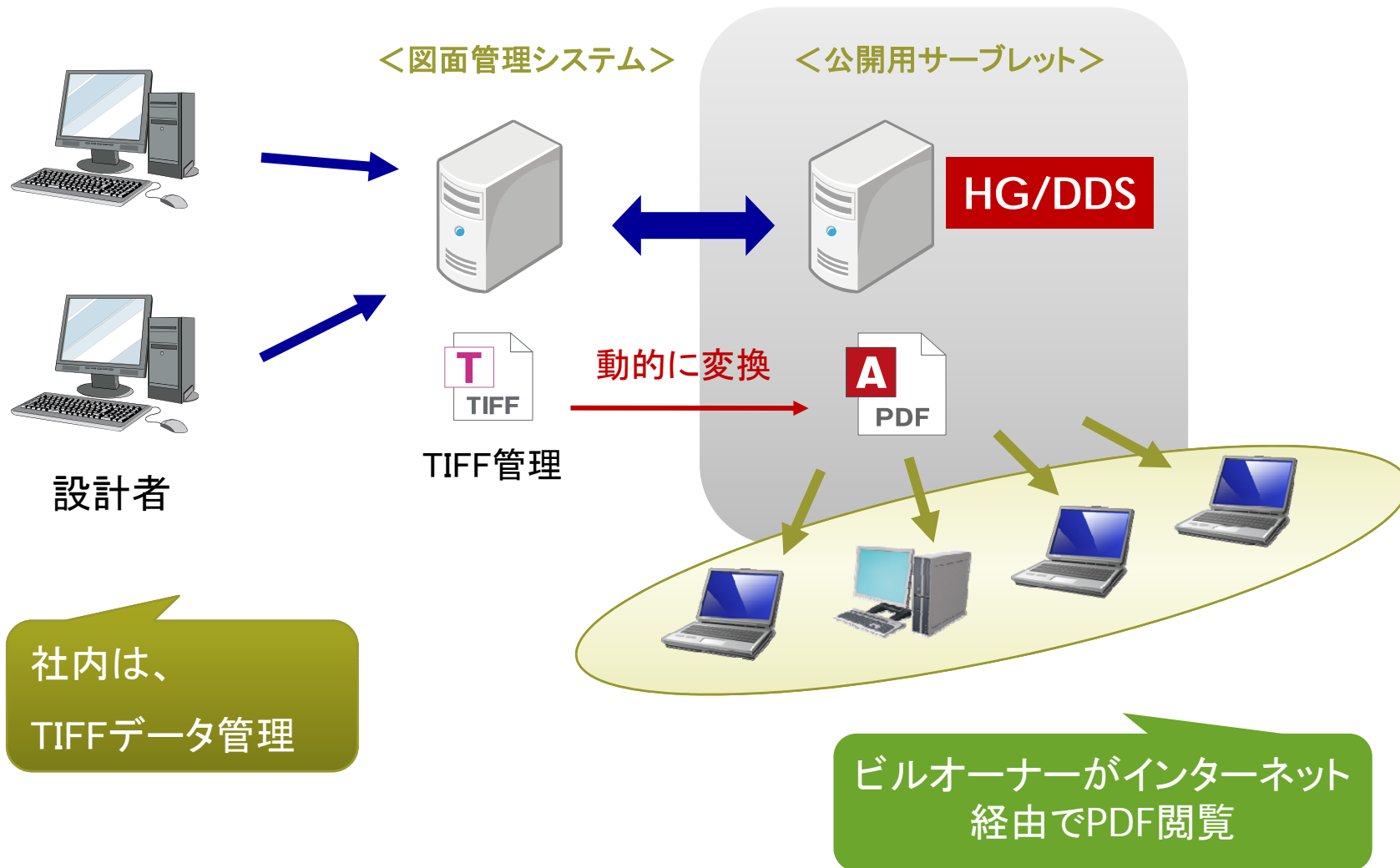
- 1) PDFで公開する事を提案
  - ・世界標準フォーマット
  - ・Acrobat Readerでの閲覧により、TIFF表示の問題点をクリア
- 2) 図面公開システム(サーバレット)に組み込むことにより、現システム変更を最小限に。かつ、データの2重化を防ぐ提案を実施
  - ・高速にPDF生成する
  - ・java Beans™コンポーネントで提供
  - ・セキュリティ設定に対応

## 成果

HG/DDS  
+  
公開用サーバレット

- ・データの2重化なしで、安定したPDFで公開。現システムの変更を最小限で可能にした。
- ・安価なシステム構築
- ・ファイル単位でのセキュリティ構築

# 大手ゼネコンシステム構成図



## 社内TIFF管理している図面を、高圧縮PDFにして配信

### 背景

#### ●課題

- 1)図面や関係資料の電子配布を進め、即時性と紙配布コストを減らす。
- 2)資料に署名を付け、出図先を明確にしたい。
- 3)データサイズを小さくしネットワーク負荷の減少を図りたい。

#### ●問題点

- 1)TIFF形式では、セキュリティが不安。PDFにしたい。
- 2)図面配信システムに組み込んで使用したい。

### 提案

海外のディーラへの図面配信システムをサーバレットを使用して作成している。保管図面は、TIFF形式で統一され、個々のディーラへの署名(赤字画像データ)をして配布しているが、このサーバレットに組み込めるHG/DDSでご提案。

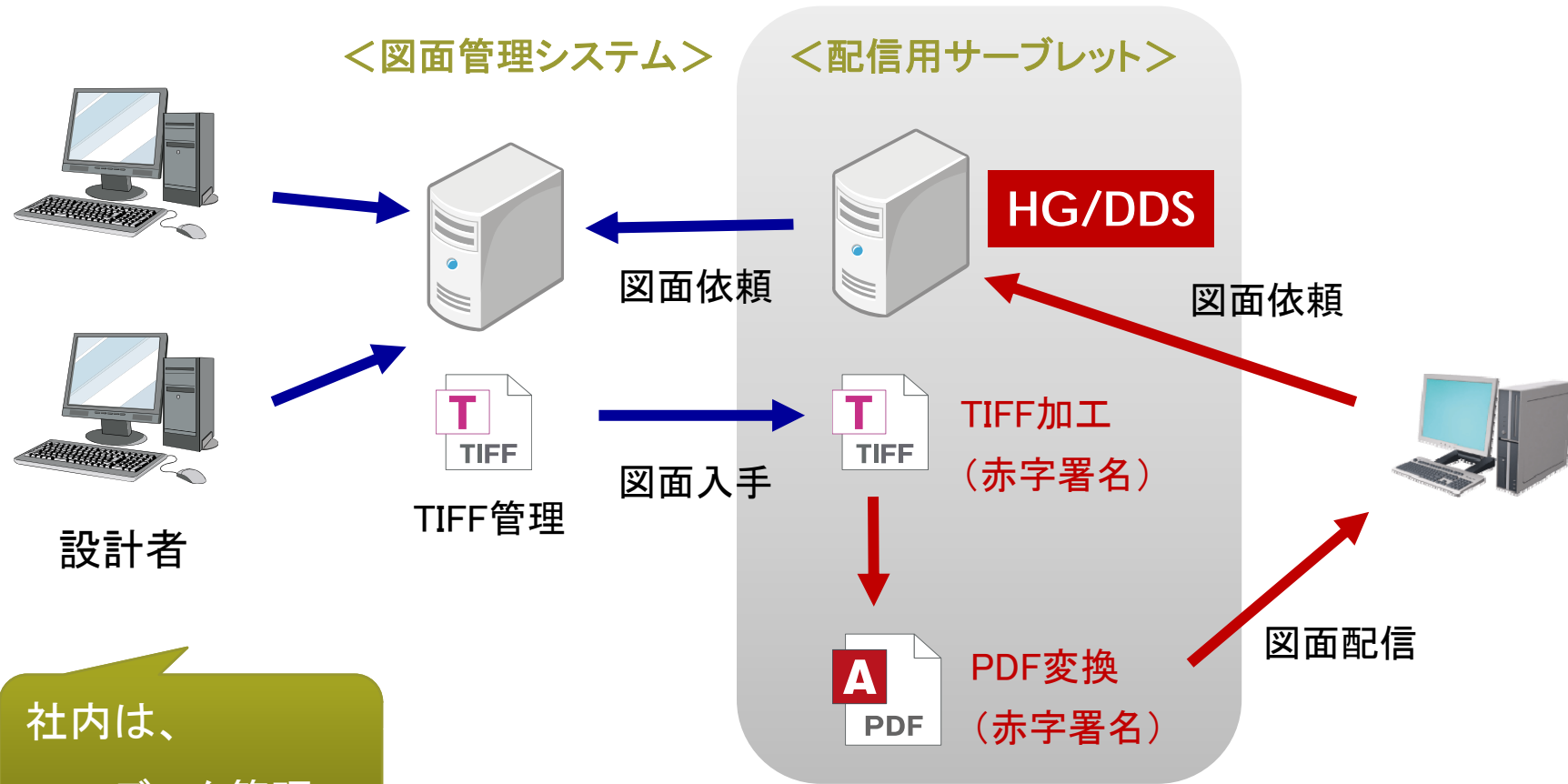
- ・java Beans™コンポーネントで提供
- ・セキュリティ設定に対応
- ・MC-PDF(高圧縮2色PDF)に対応

### 成果

HG/DDS  
+  
配信用サーバレット

- ・高圧縮PDFでデータ配信  
署名も赤字で公開可能
- ・ネットワークの負荷減少
- ・現システムの変更を  
最小限で可能にした。

# 工作機器製造業 システム構成図



社内は、  
TIFFデータ管理

- ・MC-PDF(高圧縮2色PDF)
- ・印刷/ダウンロード禁止セキュリティ