

3ヶ月でできる Linux & Java の基幹システム

2005年8月4日

ゼネラル・ビジネス・サービス株式会社
システム・サービス事業部

稲葉 達哉

Agenda

- はじめに(事例「IBMユーザー研究会連合会」の概要)
- インフラ環境になぜLinuxを採用？
- システム構成
- アプリケーション開発
- 今後の展開

IBMユーザー研究会連合会とは？

- IBMユーザー研究会は、IBMシステムを使用もしくは使用予定の企業、官公庁、学校などが加入しているコンピューター利用者の団体で、現在、全国16地区でそれぞれのIBMユーザー研究会が活動を展開しています。IBMシステムの有効利用についての研究・発表を行ない、情報交換を通じて知識と経験の向上を目指します。あわせて各種の活動を主催し、会員相互の利益と親睦を図ります。
- 各地区のIBMユーザー研究会は、年1回の総会をはじめ、例会、セミナー、研究会、企業見学会、レクリエーション・プログラムなど、会員相互の親睦や情報交換のためのプログラムを、それぞれの地区の特徴を活かして開催します。

お客様の要望/問題点

– 開発期間

- 短期間での開発（2003年 3月-5月）

– 開発体制

- GBS社内で要員計画

– スキル

- Java

- Webアプリケーション開発
- 以前のシステムがNotesであったため、すべて書き換えが必要

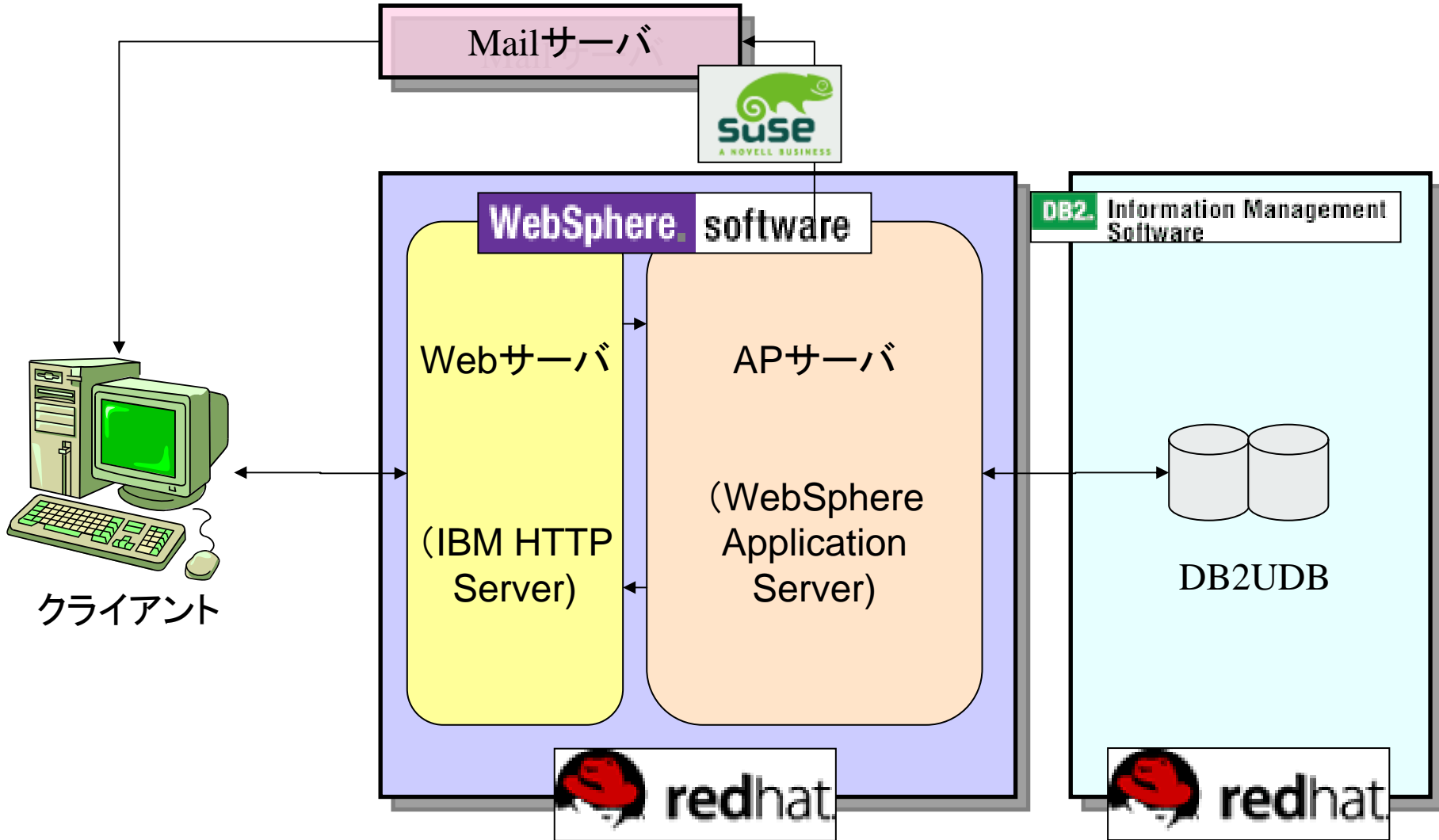
- Notes & DB2

- 以前のシステムの分析
- 以前のデータ移行

インフラ環境になぜLinuxを採用？

- Linuxを採用した理由
 - ユーザー要件
 - 担当者のスキル
 - コストパフォーマンスが良い
 - ウィルスによるセキュリティ対策
 - IBMミドルウェアの採用
 - Linuxとの親和性の高さ
 - DBMS
 - » DB2UDB 7.2 Enterprise Edition
 - Webアプリケーションサーバ
 - » WebSphere Application Server 4.0 Advanced Edition

OS (Linux) 上でのシステム構成



開発に当たっての要件

- 短期間でコアな部分を稼動
 - J2EEフレームワークを採用
 - GBS独自開発のフレームワークを採用
 - できるだけビジネスロジックに集中できるように
 - 機能ごとにサブシステム化し、並行して反復的に開発
 - いち早く動作するものを開発
 - 反復的な開発により、品質を向上

開発スケジュール

		1ヶ月				2ヶ月				3ヶ月				4ヶ月				5ヶ月				
システム名		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	Week
1	サブシステム1	■	■	■	■																	
2	サブシステム2			■	■	■	■	■	■													
3	サブシステム3							■	■	■	■	■	■									
4	サブシステム4											■	■	■	■	■	■					

開発スケジュール

		1ヶ月				2ヶ月				3ヶ月				4ヶ月				5ヶ月				
システム名		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	Week
1	サブシステム1	■	■	■	■																	
2	サブシステム2			■	■	■	■	■	■													
3	サブシステム3							■	■	■	■	■	■									
4	サブシステム4											■	■	■	■	■	■					

コアなサブシステム

開発スケジュール

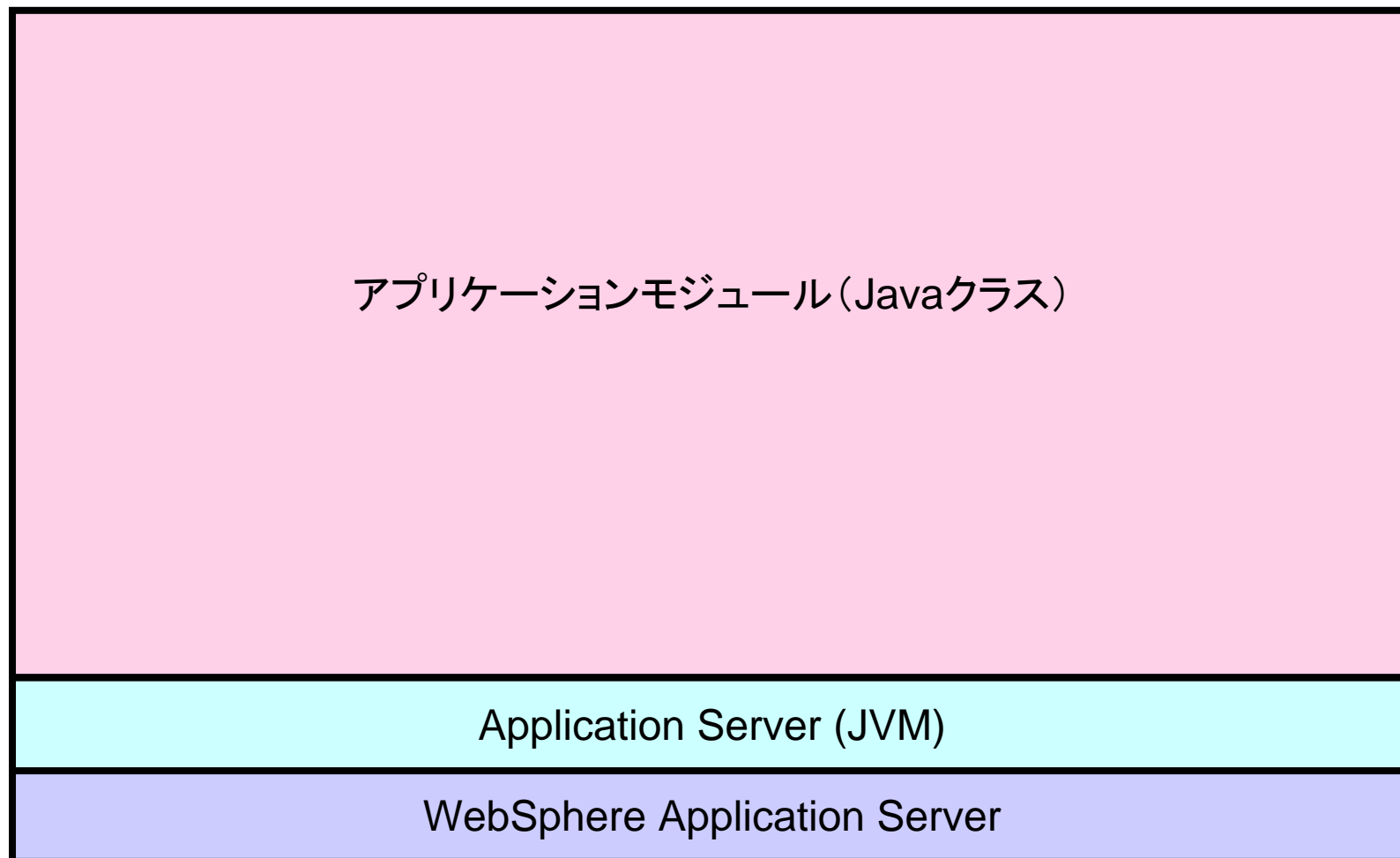
		1ヶ月				2ヶ月				3ヶ月				4ヶ月				5ヶ月				
システム名		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	Week
1	サブシステム1	■	■	■	■																	
2	サブシステム2			■	■	■	■	■	■													
3	サブシステム3							■	■	■	■	■	■									
4	サブシステム4											■	■	■	■	■	■					

J2EEフレームワークを採用

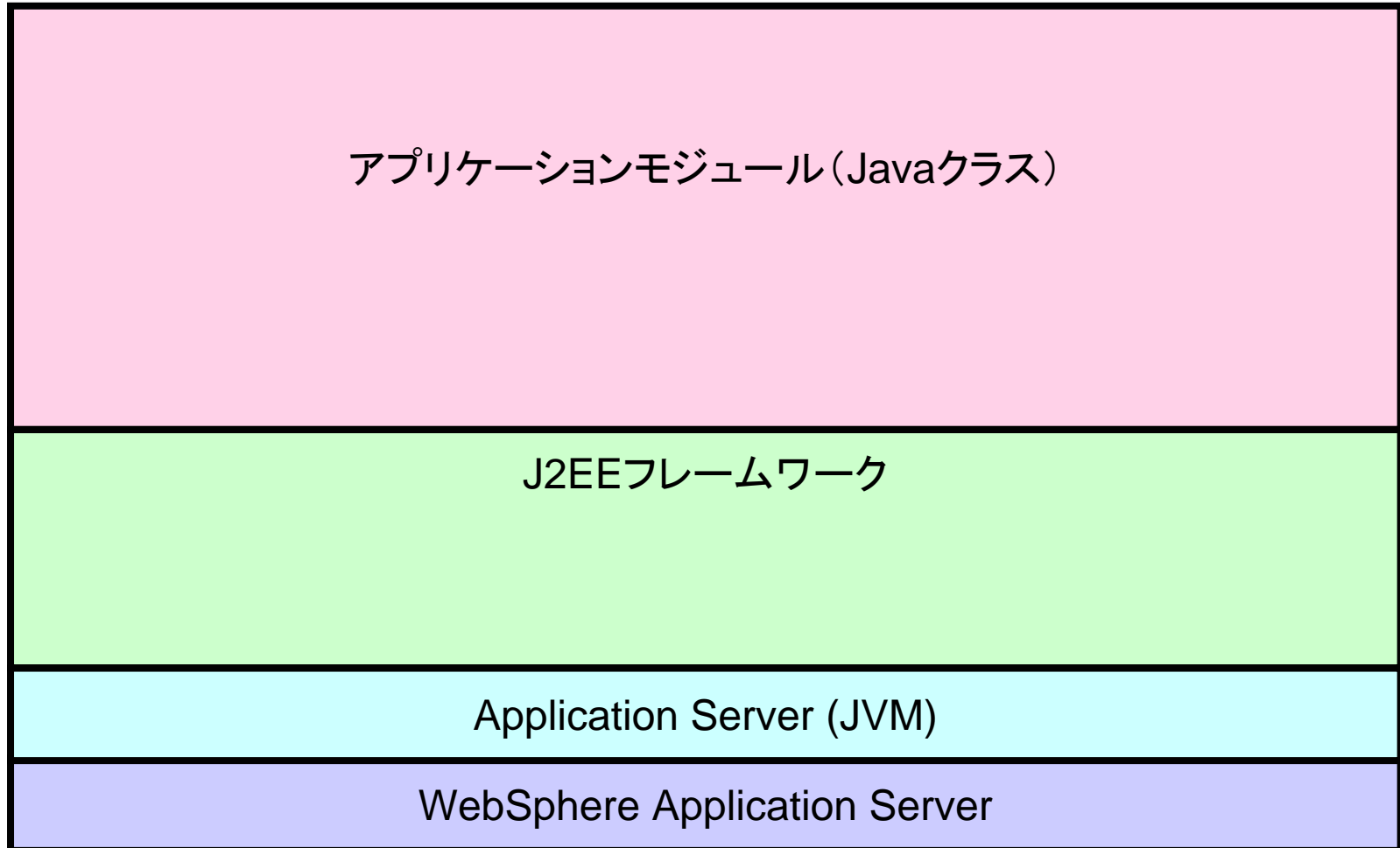
- GBS独自開発のJ2EEフレームワークを採用
 - オープンなJ2EEフレームワーク「Struts」との比較
 - 比較要素「ビジネスロジック」のフレームワーク

- 共通の機能をフレームワーク化
 - 検索ロジックの簡素化
 - 検索結果の表示ルール(表示件数やボタン)の一元管理
 - ダウンロードロジック

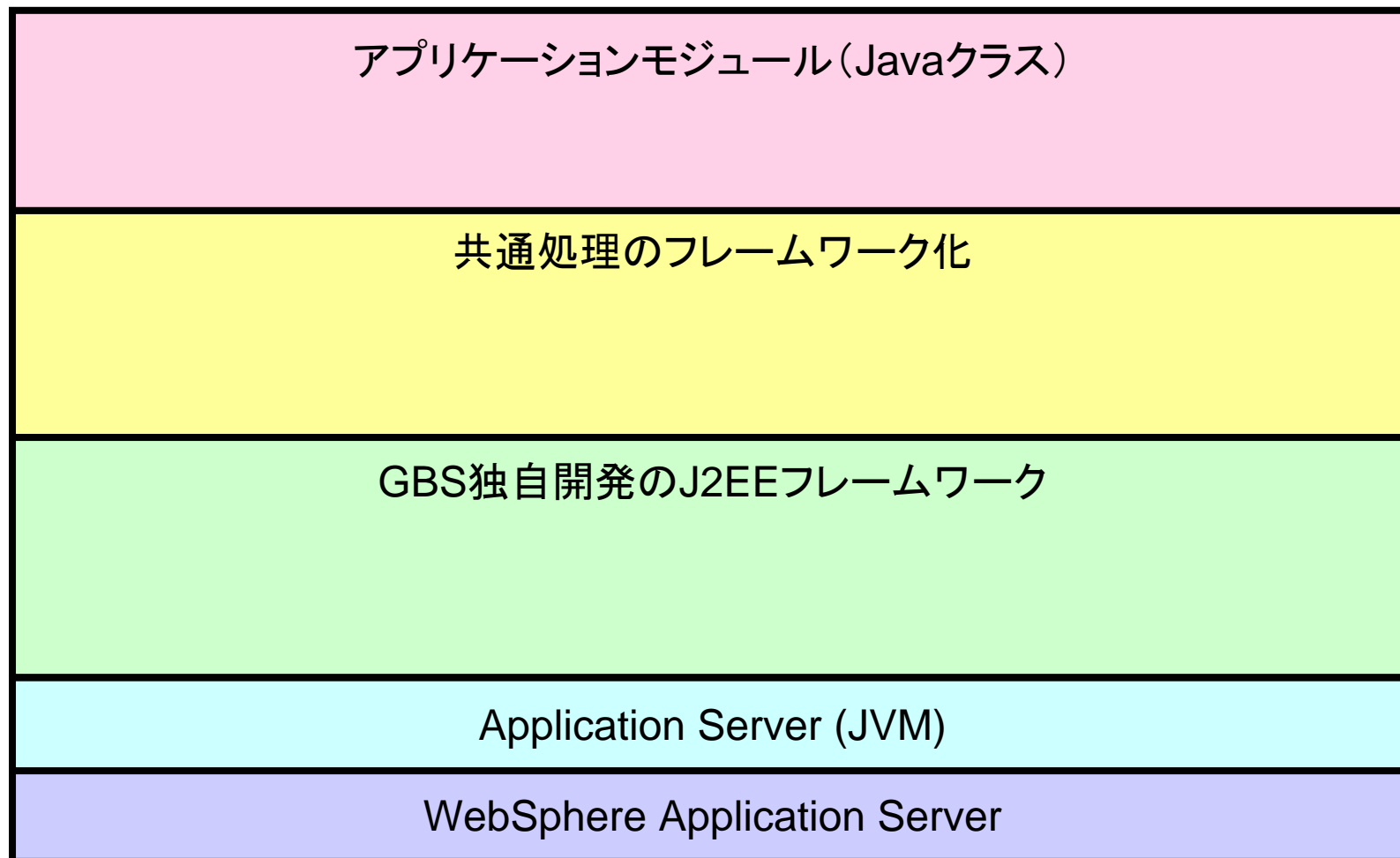
アプリケーションの層構造 (1) ～シンプルな例～



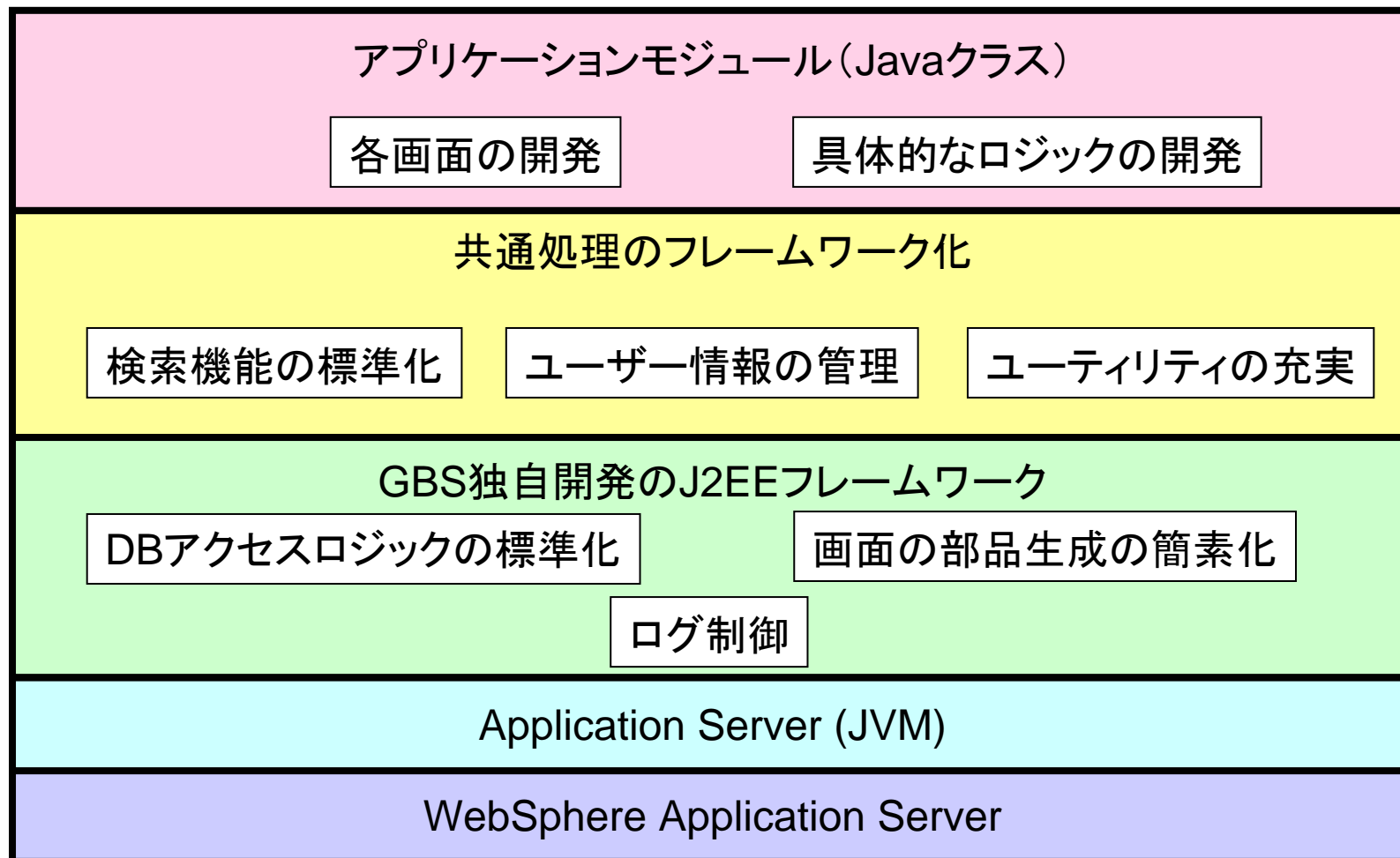
アプリケーションの層構造 (2) ～一般的な例～



アプリケーションの層構造 (3) ～今回の例～



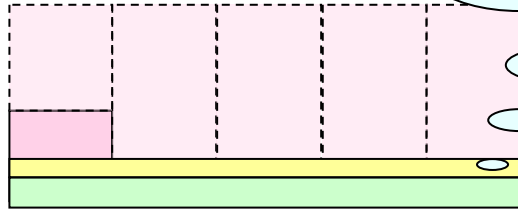
アプリケーションの層構造 (3) ～今回の例～



並行して反復的に開発

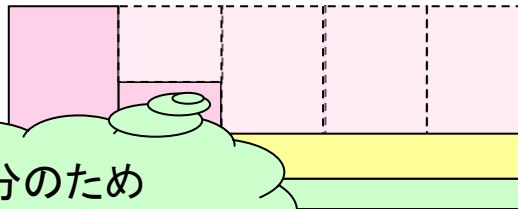
● 今回の開発例

– サブシステム1での開発



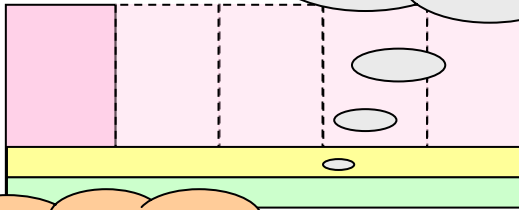
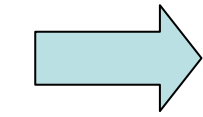
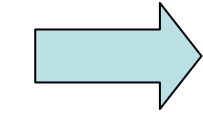
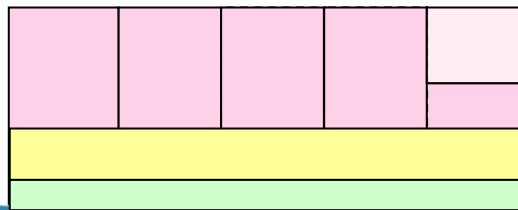
プロトタイプ開発で
共通部分を開発

– サブシステム2での開発

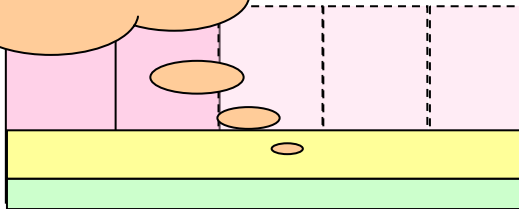


共通部分のため
開発効率が向上

– サブシステムnでの開発

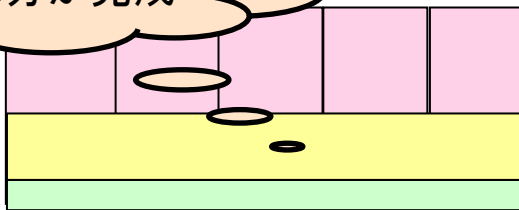


完成段階で共通
部分の抜き出し



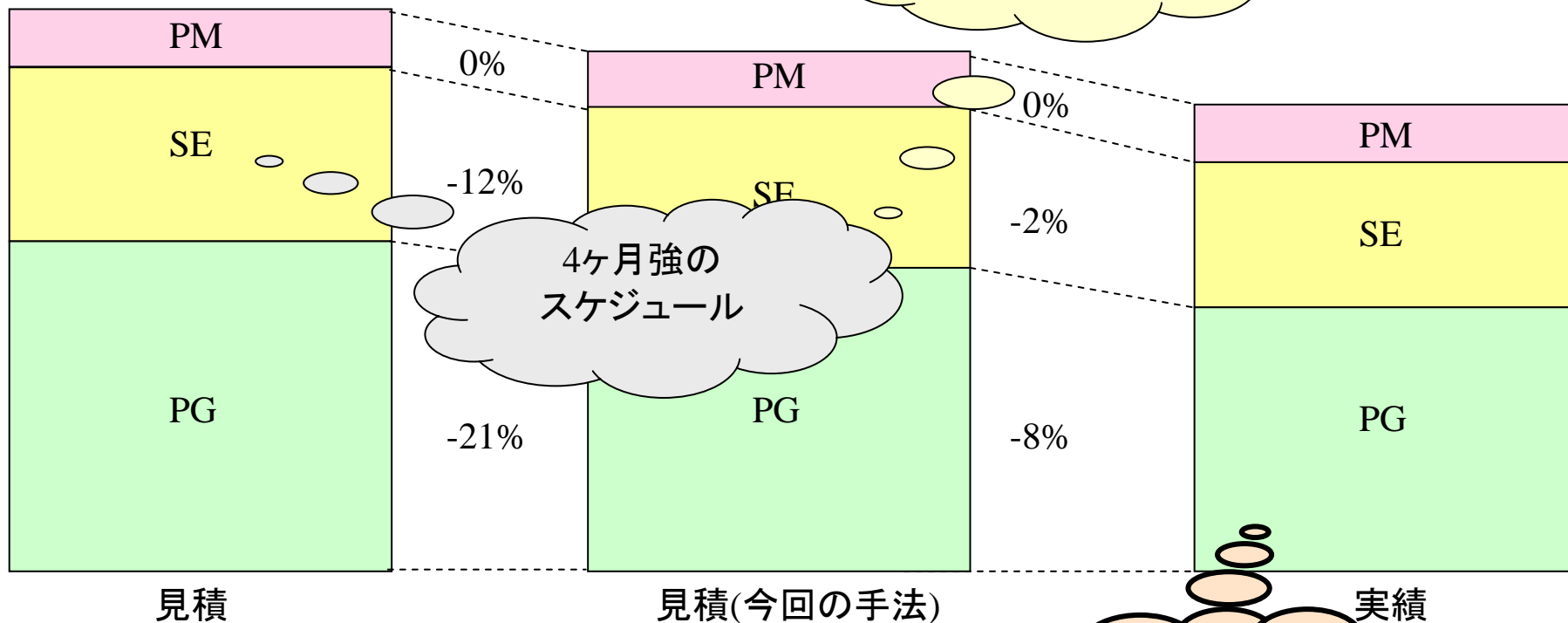
共通部分がほぼ
完成

共通部分が完成



開発を終えての成果

ワークロードを分析



最終的に
開発工数削減に
成功!

今後の展開

- ツールを利用し効率よく開発
 - 簡単Webアプリケーション開発ツールの利用
 - JSF
 - adieu (SOA準拠ソリューション)

IBM alphaWorksサイトよりダウンロードが可能
(<http://www.alphaworks.ibm.com/tech/adieu>)

ご清聴ありがとうございました。