

# SakaiとLTI連携する Java自習システム

法政大学  
情報メディア教育研究センター

藤井 聡一郎

- ・ Sakaiと連携して動作するJava学習支援システムの開発
  - ・ ソースコードを提出させ、課題の仕様を満たすかを自動チェック
  - ・ ユーザ間のコミュニケーション機能
    - ・ 教員による進捗確認、学生同士の回答閲覧
- ・ Sakaiとの連携にはLTIを使用
  - ・ 開発や保守・運用のしやすさ
  - ・ Sakai以外のCMSとも連携が可能
- ・ 来年度の春学期から運用予定

- 
- ・ IT基礎教育の授業の増加
    - ・ 多くの学部でExcelやプログラミングなどの授業を実施
  - ・ CMSでのレポート機能の使用頻度は高い

Excel	270
HTML	113
Java	105 (JavaScriptを減算したもの)
PHP	66 (誤差多め)
C言語	56
JavaScript	34
C++	32
Visual Basic	32
Ruby	12
Python	6
Perl	4
C#	2
Action Script	1

IT基礎教育科目は  
多くの学部で実施されている

これらの授業を実施している主な学部・研究科

国際文化学部、デザイン工学部、理工学研究科、経営学部、工学研究科、情報科学研究科、理工学部、情報科学部、イノベーション・マネジメント専攻、社会学部、リベラルアーツ、デザイン工学研究科

- ・ CMSと連携したIT基礎教育のための課題提出ツールの開発
  - ・ 当面の対象はJavaプログラミング
- ・ 回答の自動チェックやユーザ間のコミュニケーション機能により教育の質の向上を目指す
  - ・ 自動チェックを導入し課題の効率化
  - ・ ユーザ間のコミュニケーション機能
    - ・ きめ細やかな学生へのサポート
    - ・ 学習のモチベーション向上
- ・ 将来的には
  - ・ プログラミング以外のコンテンツへの対応
  - ・ モバイル端末, tegakiプロジェクト, ラーニングアナリティクスなどとの連携

- 
- ・ システムの提供する機能
    - ・ プログラミング学習用コンテンツ
      - ・ 回答の自動チェック
    - ・ ユーザ間のコミュニケーション機能
  - ・ CMSとの連携
    - ・ LTIを利用
      - ・ CMSとは別サーバでの稼働

- 
- ・ 主なコンテンツ
    - ・ 授業補助コンテンツ
    - ・ 自習用コンテンツ
    - ・ 期末試験用コンテンツ
  - ・ 問題形式
    - ・ 選択, 穴埋め
    - ・ プログラムのソースコードを提出
  - ・ 回答の自動チェック
    - ・ 入出力のペアによるチェック
    - ・ JUnit
    - ・ ASTを用いた静的解析
      - ・ AST Abstract Syntax Tree 抽象構文木

- ・ コミュニケーション機能
  - ・ 学生の進捗可視化
    - ・ 教員向け
    - ・ 全体の進捗の把握に利用
  - ・ 回答の相互閲覧
    - ・ 学生向け
    - ・ モチベーションの向上
      - ・ 自分とは異なる解法
      - ・ 公開することによる盗作防止

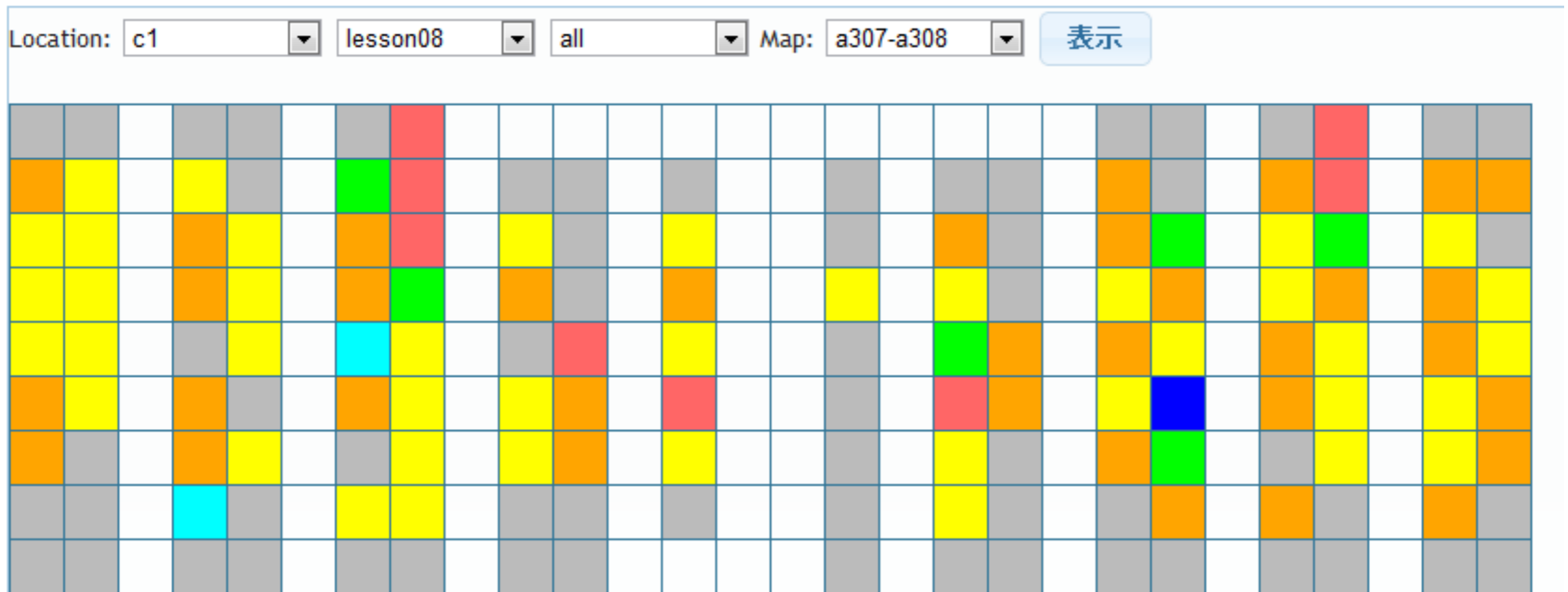


## Classroom Map

学生の進捗状況によって座席を色分けします。5分ごとに自動更新されます。

色分けの意味は以下のとおり。

空席 未完了 レベル1完了 レベル2完了 レベル3完了 レベル4完了 レベル5完了



## exercise09:家系図(9)

問題

提出履歴

他の人の提出を見る

Twitter

全11件 投稿時間が新しい順  最後に投稿されたものだけ表示  完了しているものだけ表示

<p>10/12/10 14:06:05 fuji [redacted]</p> <pre>#include &lt;stdio.h&gt;  struct person {     int age;     char name[20];     struct person *father;     struct person *mother; };  void print_family(struct person *p) {     int i, j, k, l, m;     struct person *p1, *p2, *p3, *p4, *p5;      printf("%s\n", p-&gt;name);     for (i = 0; i &lt; 2; i++) {         p1 = p;         if (i == 0 &amp;&amp; p-&gt;father != NULL) {             p = p-&gt;father;             printf("%s\n", p-&gt;name);         } else if (i == 1 &amp;&amp; p-&gt;mother != NULL) {             p = p-&gt;mother;             printf("%s\n", p-&gt;name);         }     }     for (j = 0; j &lt; 2; j++) {</pre> <p>views: 0 comments:</p> <p>閉</p>	<p>10/12/07 15:54:22 rotartsinimda [redacted]</p> <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt;  struct person {     char name[20];     int age;     struct person *father;     struct person *mother; };  void print_name(struct person *p) {     printf("%s", p-&gt;name); }  void set_name(struct person *p, char name[]) {     int i;     i = 0;     while (name[i] != 0) {         p-&gt;name[i] = name[i];         i++;     }     p-&gt;name[i] = 0; }</pre> <p>views: 0 comments:</p> <p>閉</p>	<p>10/12/07 15:36:17 [redacted]</p> <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt;  struct person {     int age;     char name[20];     struct person *father;     struct person *mother; };  void set_name(struct person *p, char name[]) {     int i;     i = 0;     while (name[i] != 0) {         p-&gt;name[i] = name[i];         i++;     }     p-&gt;name[i] = 0; }  void print_name(struct person *p) {     if(p==NULL)         printf("unknown\n");     else         printf("%s\n", p-&gt;name);</pre> <p>views: 1 comments:</p> <p>閉</p>
--	---	---

<< < 1 2 3 4 > >>

---

- ・ CMSとの連携手法

1. CMSの提供する独自の拡張機能の利用

- ・ ex) Sakaiのtool (MAX/C), maharaのプラグイン

2. LTIを用いた連携

- ・ メリット

- ・ CMSの提供するAPIなどの利用が可能
- ・ ユーザ情報の取得や他の機能との連携が容易

- ・ デメリット

- ・ 特定のCMSでしか利用できない
- ・ 基本的にCMSと同じサーバ上で動作する
  - ・ 組み込んだツールによってシステム全体に障害が発生することもある
- ・ 開発環境はCMSによって依存する
  - ・ 使用言語やフレームワークなど

- ・ LTI: Learning Tools Interoperability
  - ・ 多くのCMSなどで採用されている機能追加のための仕様
- ・ 複数のバージョンがある
  - ・ LTI1.0, 1.1, 1.2, 2.0
- ・ CMS(LTI Consumer)にLTIに対応したツール(LTI Provider)を設定する
  - ・ Consumerへの追加は基本的にパラメータの設定のみ
- ・ ConsumerとProviderは別々のサーバ上で動作が可能

- ・ メリット

- ・ LTIに対応した多くのシステムへの組み込みが可能
- ・ CMSへの追加のしやすさ
- ・ CMSとは別のサーバで稼働する
  - ・ 保守, 管理, 開発がしやすい
  - ・ 開発もCMSで使用されている言語やフレームワークに依存しない

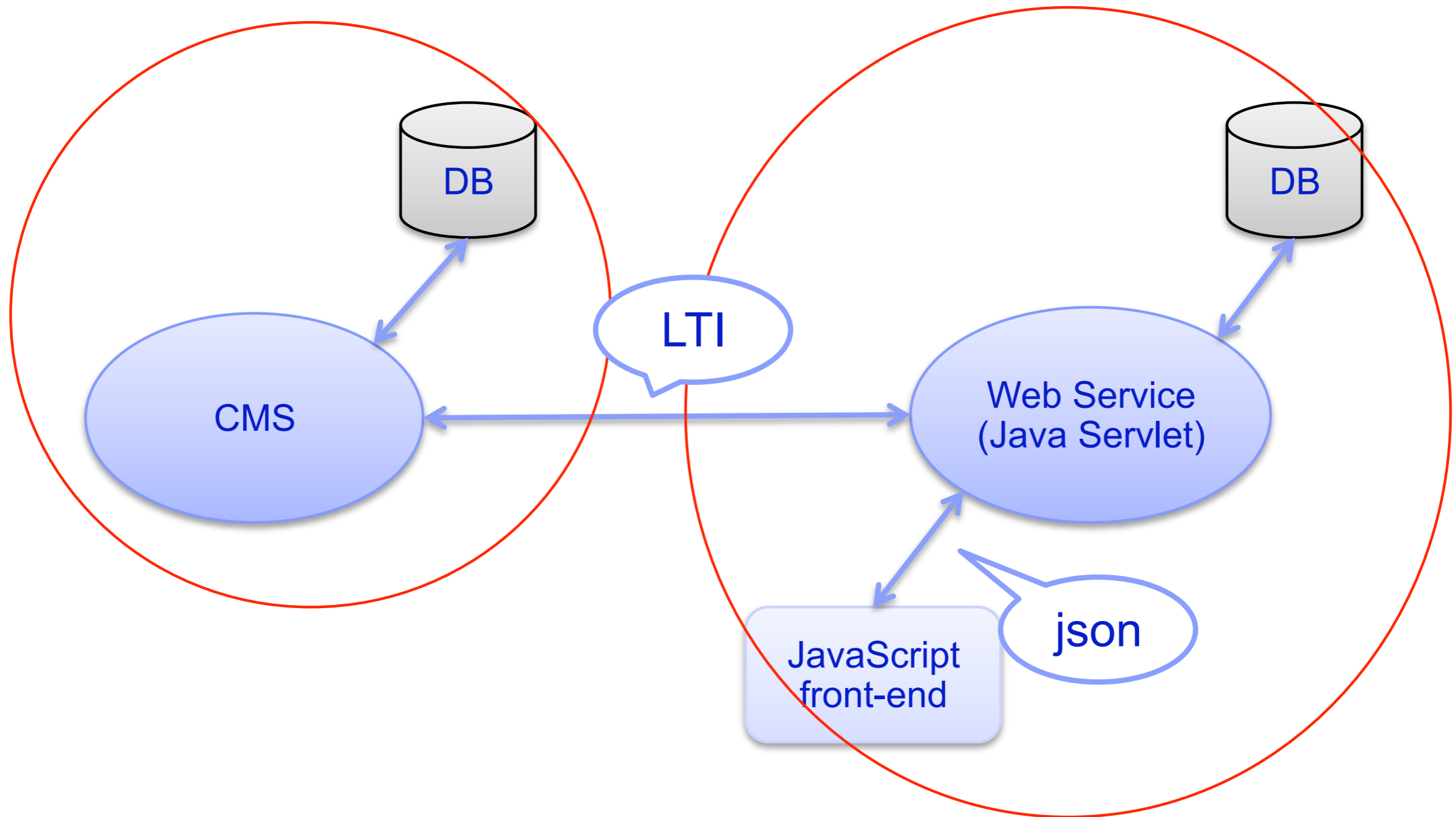
- ・ デメリット

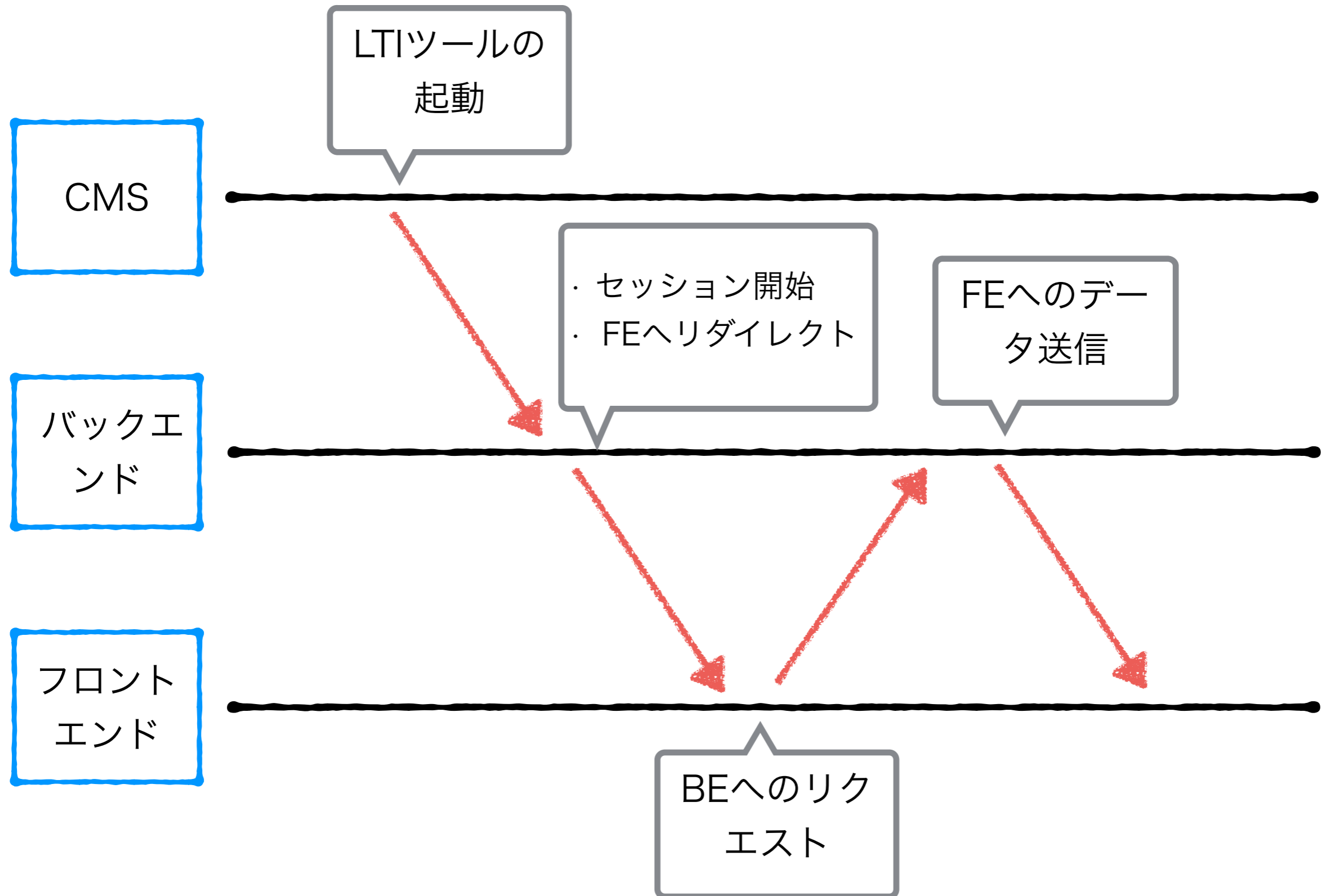
- ・ CMSとのデータのやり取りが難しい
  - ・ CMSの提供するAPIなどの利用は困難
    - ・ LTI2.0では可能?

- 
- ・ 今回はLTI1.0を採用する予定
    - ・ 機能は少ないが、すでに多くのシステムで採用されている
  - ・ 将来的な2.0への移行も考慮

- 
- ・ バックエンド
    - ・ Javaで実装
    - ・ JAX-RS: Java API for RESTful Web Services
      - ・ 実装ライブラリはApache CXFを利用
    - ・ Hibernate
    - ・ Spring DI
  - ・ フロントエンド
    - ・ HTML + jQuery
  - ・ フロントエンドとバックエンド間はjsonで通信









- ・ LTIに対応したIT基礎教育支援ツールの作成
  - ・ 当面の対象はプログラミング教育
- ・ LTIのメリット
  - ・ CMSとの分離
    - ・ 開発・保守・運用のしやすさ
- ・ システム構成
  - ・ フロントエンド: HTML, jQuery
  - ・ バックエンド: JAX-RS (Webサービス)
    - ・ 両者はjsonで通信
- ・ 将来的には
  - ・ プログラミング教育以外のコンテンツへの対応
  - ・ tegaki, eポートフォリオ, ラーニングアナリティクスとの連携

---

ご清聴ありがとうございました