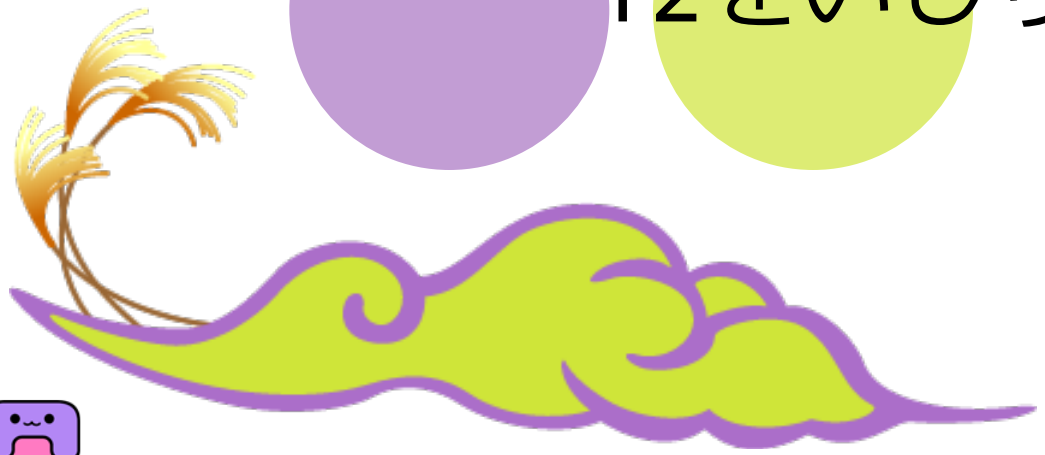
Seasar Conference
2009 Autumn

T2 Hacks

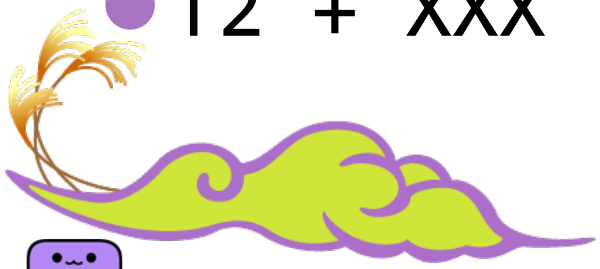
～T2をいじり倒そう2009～

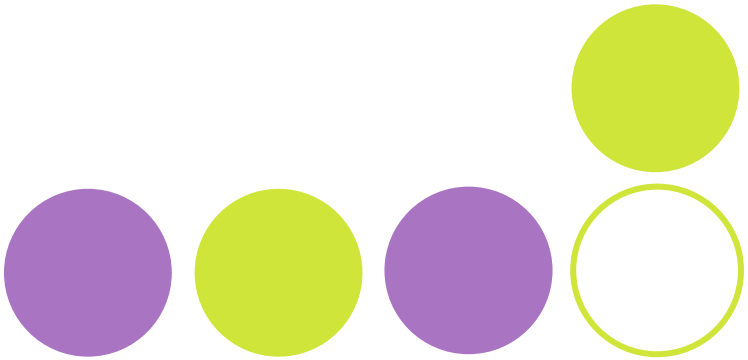


Seasar Conference 2009 Autumn

全体のAgenda

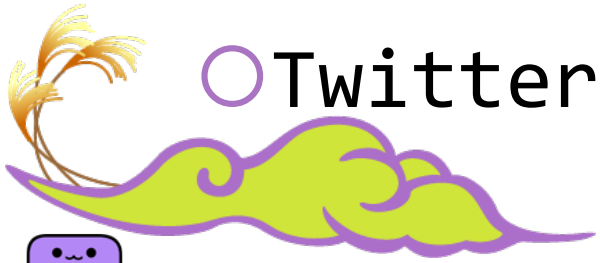
- チーム紹介
- T2を拡張する
- T2やるべ
  - EclipseプラグインViliによるT2開発事始め
- AMFで本気出す
  - T2AMFを使ってみよう
- T2 + XXX

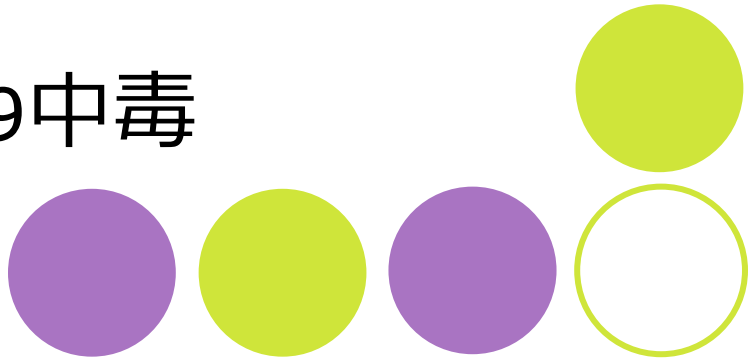




自己紹介

- 名前
  - 大谷　晋平
- ID
  - shot6
- 所属
  - ISID
  - Twitter/iPhone/DQ9中毒

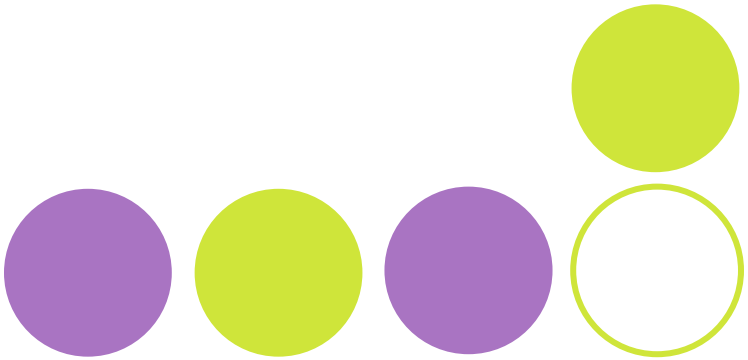




Seasar Conference 2009 Autumn

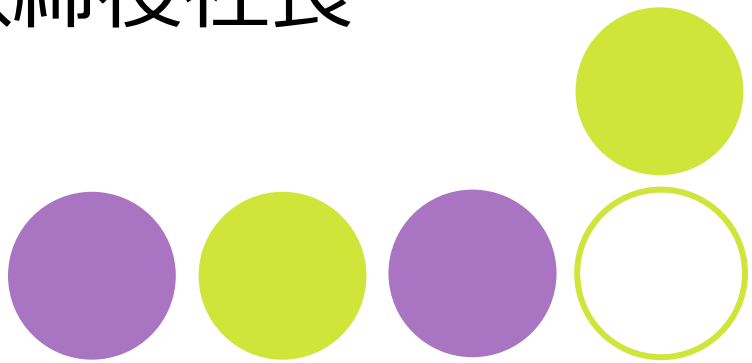
自己紹介

- 名前
  - 横田健彦
- ID
  - skirnir
- 主担当
  - Vili
- 一言
  - Ymirもやってます



自己紹介

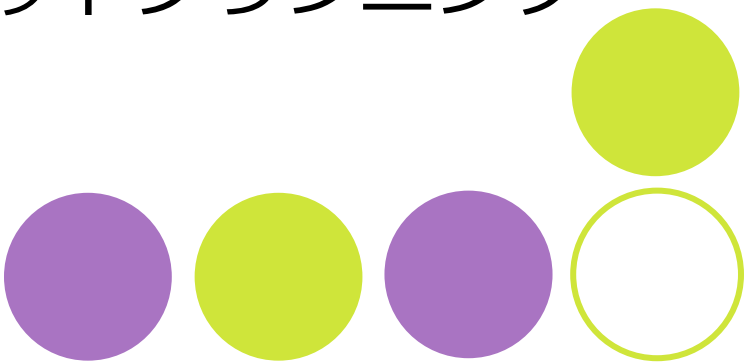
- **名前**
  - 米林　正明
- **ID**
  - yone098
- **所属**
  - 株式会社Abby　代表取締役社長



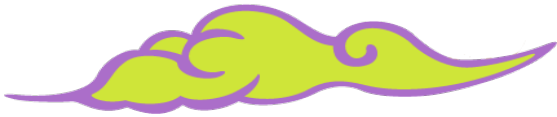
Seasar Conference 2009 Autumn

自己紹介

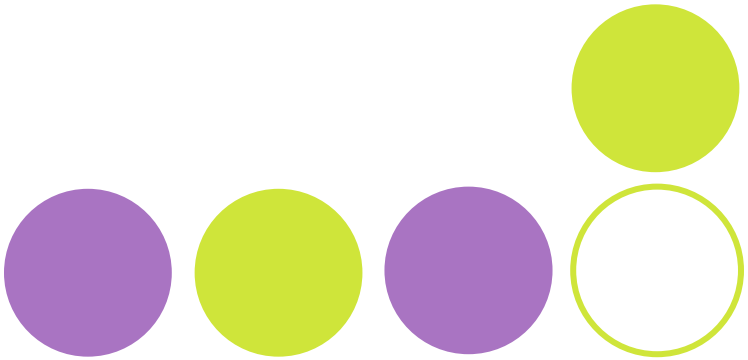
- **名前**
  - 片山　暁雄
- **ID**
  - c9katayama
- **所属**
  - 株式会社キャピタルアセットプランニング
    係長相当



Seasar Conference
2009 Autumn

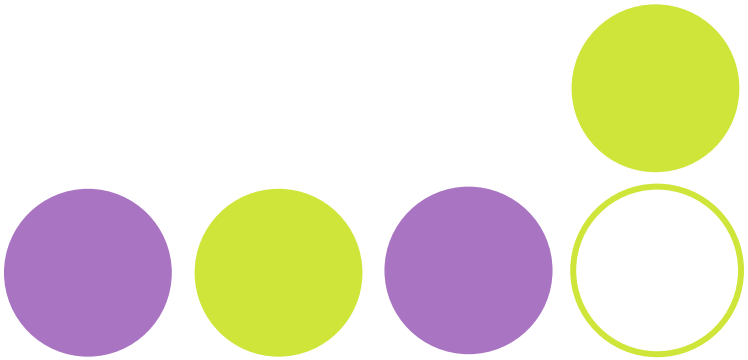


T2を拡張する

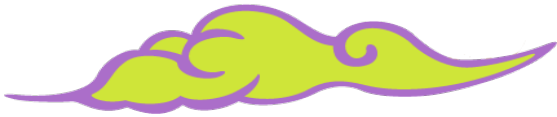


T2拡張のアジェンダ

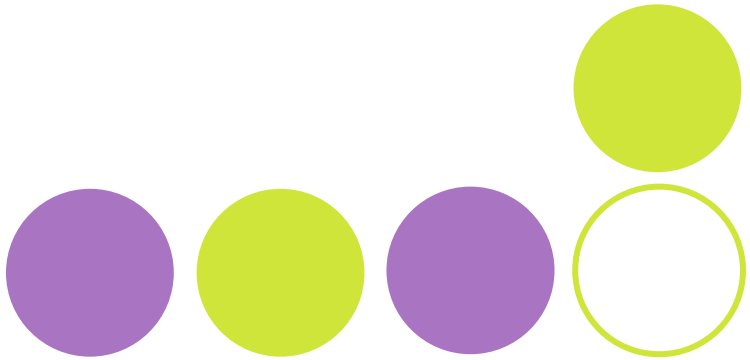
- T2について
- 拡張ポイント
- ActionInvoker
- AnnotationResolverCreator
- Plugin
- ExceptionHandler
- ContainerAdapter



Seasar Conference
2009 Autumn

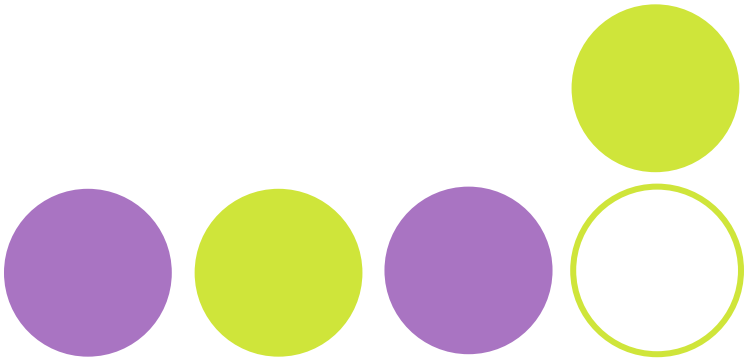


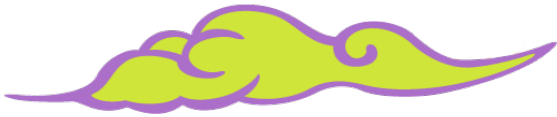
T2について



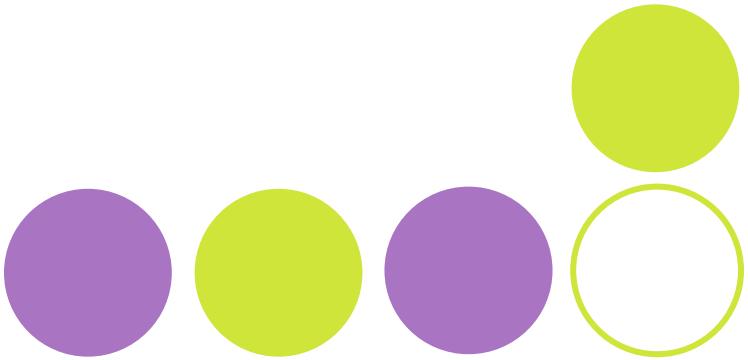
T2について

- T2について
  - 世の中を幸せにするWebフレームワーク
    - 作ってる自分も幸せ。
    - それを使ってもらえるともっとシヤワセ。
  - http://code.google.com/p/t-2/で開発中
  - 使ってる人がカスタマイズすぐに出来るのも特徴の一つ。今日はそんなとこをご紹介。



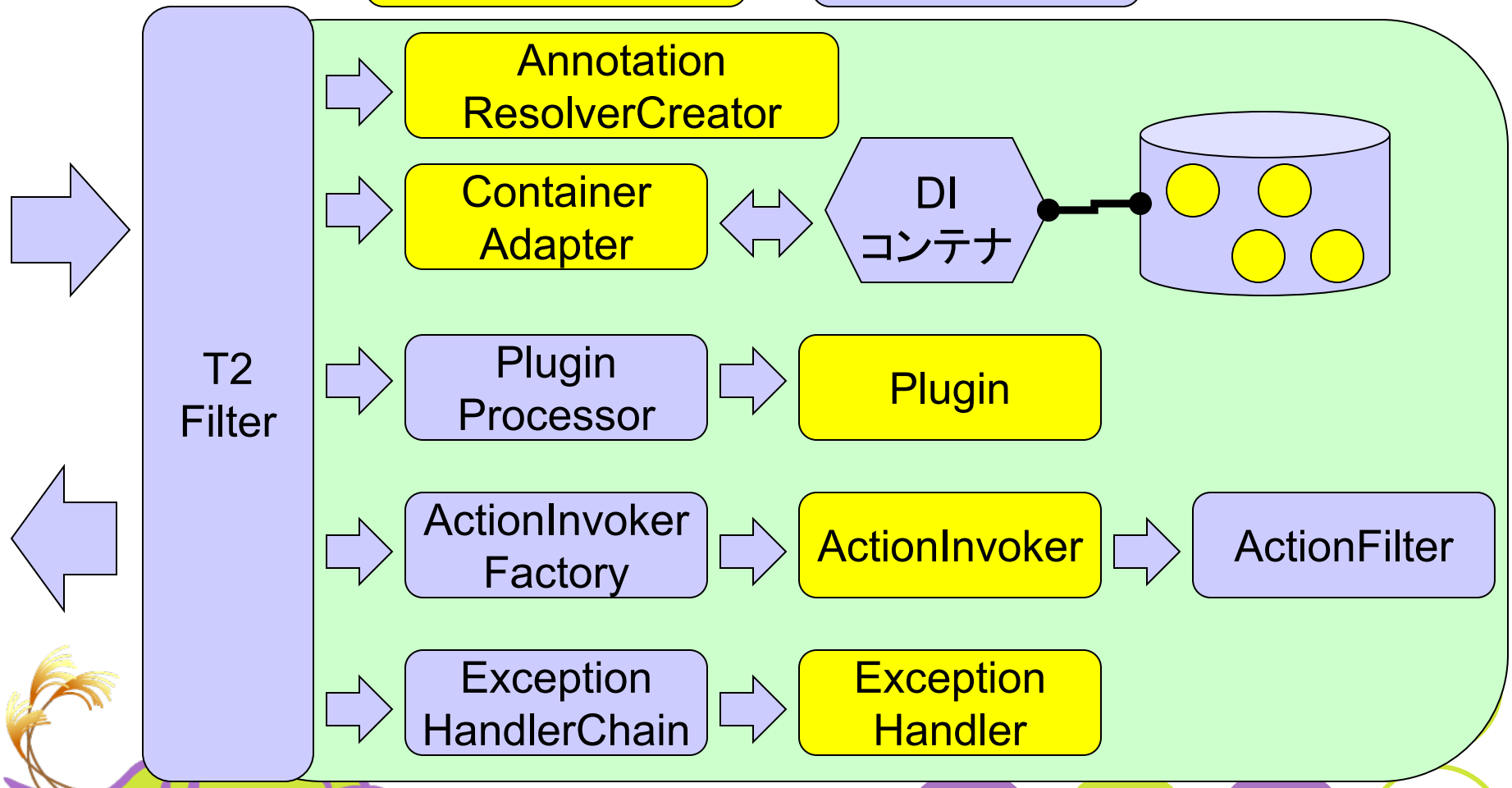


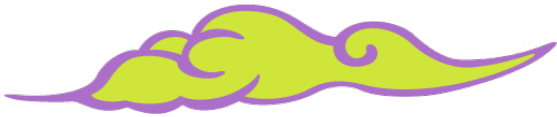
拡張ポイント



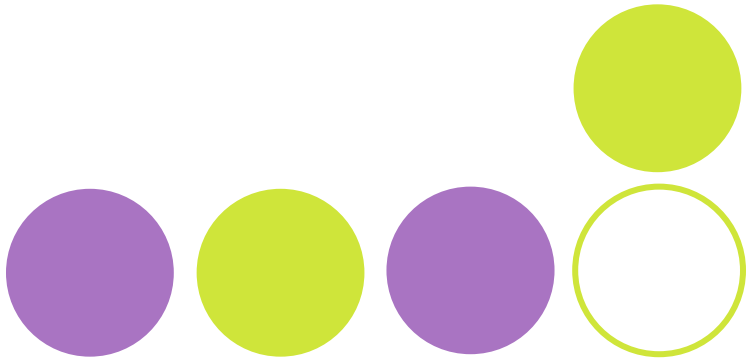
今回紹介するところ

それ以外



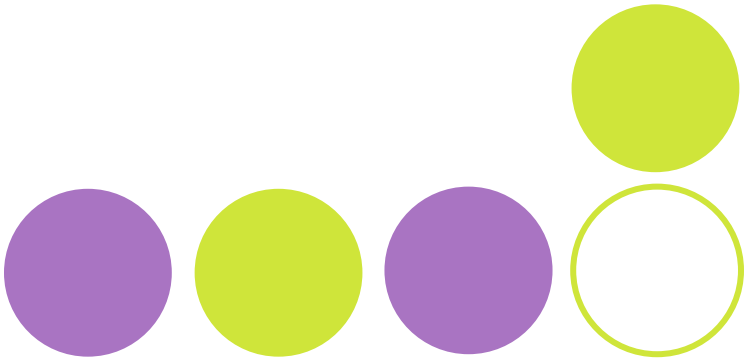


ActionInvoker



ActionInvoker

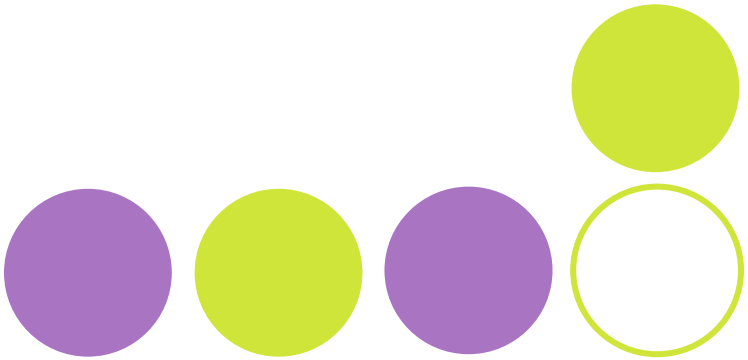
- T2のコア部分
  - Pageから適切なアクションメソッドを探し出し、実行する。
  - ActionInvoker自体も小さなFilterであるActionFilterで構成されている



ActionInvoker

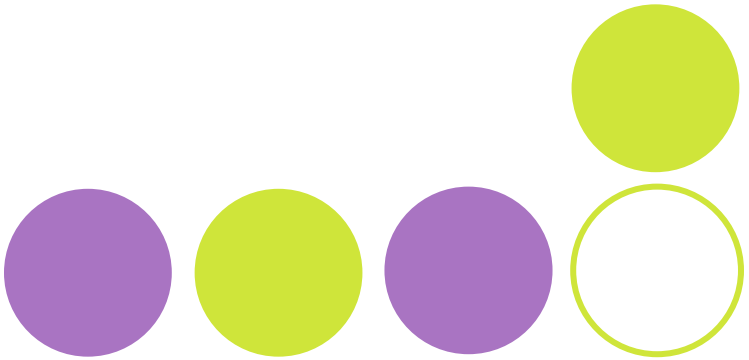
デフォルトでは、、、

```
protected ActionFilter[] filters = new ActionFilter[] {
        new ExceptionHandlerActionFilterImpl(),
        new PageCreationFilterImpl(),
        new ActionArgumentsPreparationFilterImpl(),
        new ActionInvokerFilterImpl() };
```



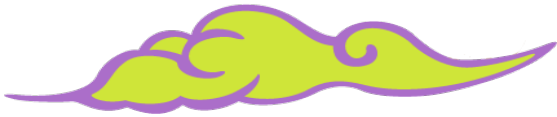
ActionInvoker

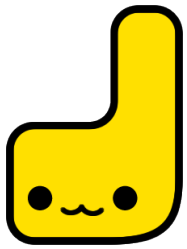
- ActionInvokerFactory
  - Factoryクラス。
  - ActionInvokerごとの実行順序を制御する
  - デフォルトはAMF用をみてから、普通のHTTPリクエスト用のをみる



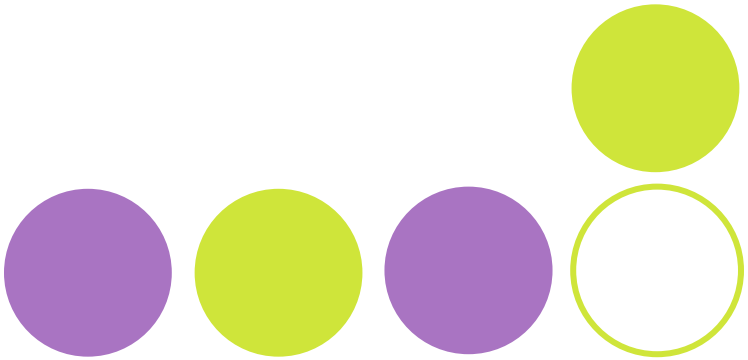
ActionInvokerの拡張

- ActionInvokerFactory
  - コンテナアダプタ経由での取得
- ActionInvoker
  - ActionInvokerFactoryからとる。
- いつ拡張すべきか
  - リクエストのフォーマットや構造が違うなどの処理方法に大幅に変更がある場合
  - ActionInvoker内のFilterの実行順序を変更したい場合



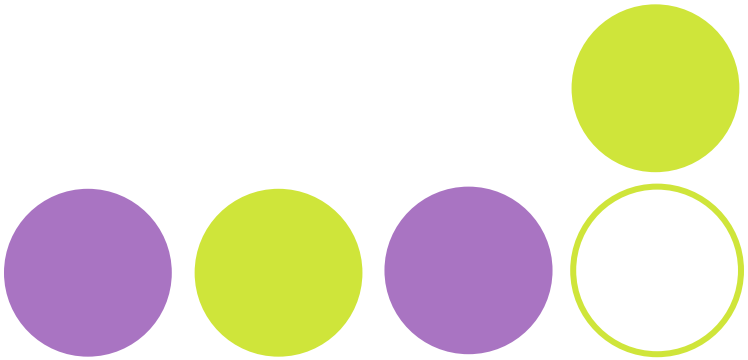


Annotation ResolverCreator



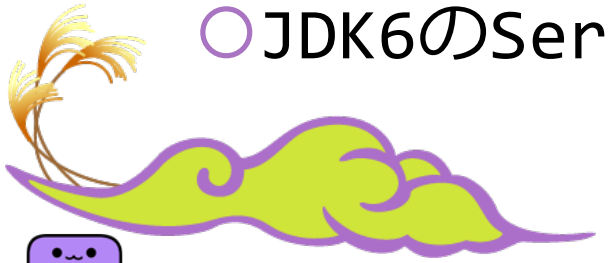
AnnotationResolverCreator

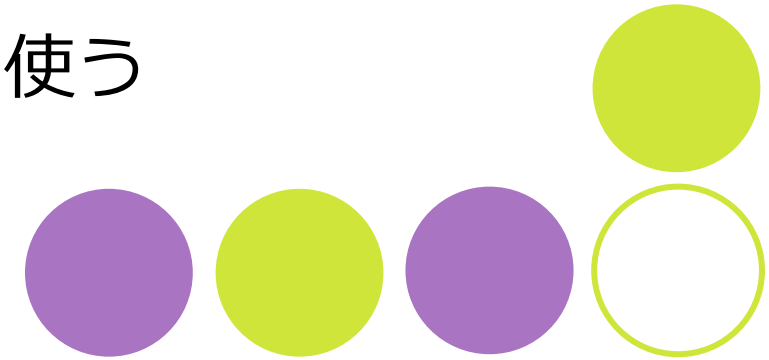
- アノテーションとハンドラの関連づけ
  - メソッドアノテーションと ActionMethodResolver
  - 引数アノテーションとParameterResolver
  - アノテーションに対応するハンドラを追加、交換、削除できる。

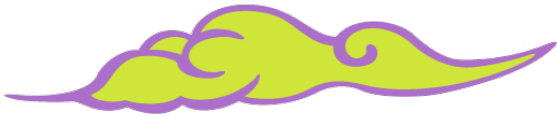


AnnotationResolverCreator

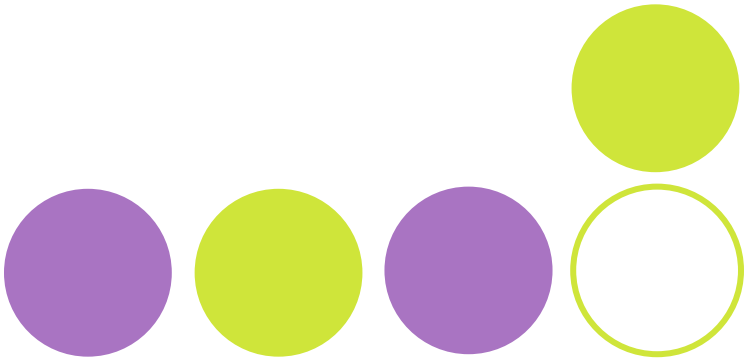
- いつ使うか
  - 自前のアノテーションとハンドラを作りたくなった場合
  - 既存の挙動が自分のケースでは上手くない場合
  - 不要アノテーションとハンドラを削りたい場合
- どのように拡張するか
  - ContainerAdapterに登録する
  - JDK6のServiceLoaderを使う





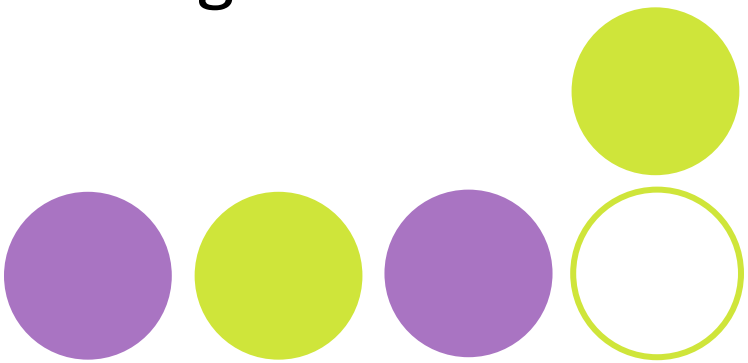


Plugin



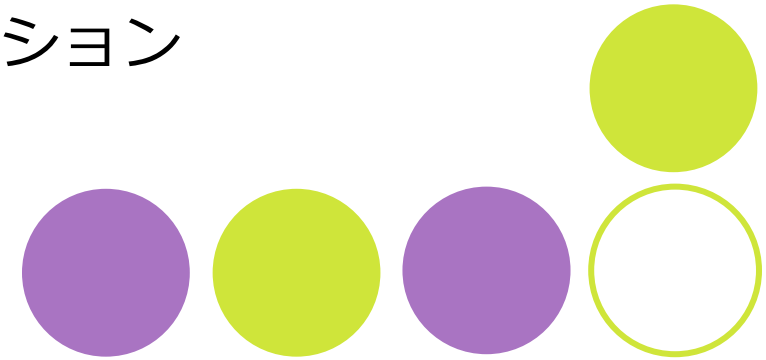
Plugin

- T2のフェーズに処理を差し込む機構
  - initialize
  - beginRequestProcessing
  - createRequest/createResponse
  - componentCreated
  - beforeActionInvoke/afterActionInvoke
  - beforeNavigation/afterNavigation
  - endRequestProcessing
  - destroy



Plugin

- どのように差し込むか
  - ContainerAdapterに定義
  - ただしDIコンテナが複数コンポーネントを返せないものの場合が現在課題
- いつPluginを作るか
  - 何か汎用的にやりたくなったら、まずPlugin化を考えてみる。
    - 事前・事後の共通バリデーション
    - 認証チェック

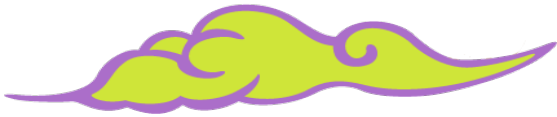


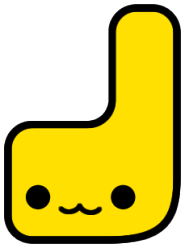
Plugin実例

- IE8の暫定対応するPlugin
  - X-UA-Compatible=IE=EmulateIE7とか

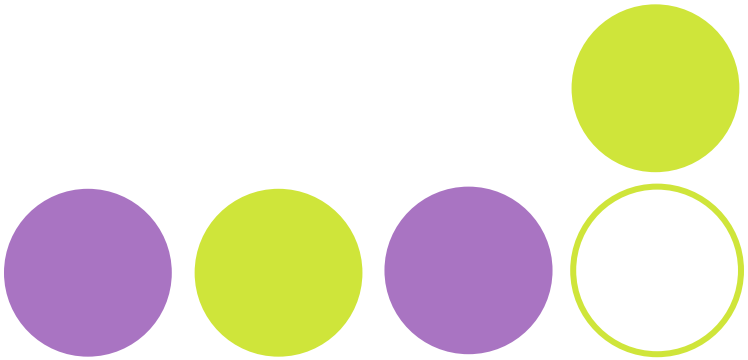
```
public class BrowserPluginImpl extends AbstractPlugin {
   protected Properties browserProperties = new Properties();
  //initでpropertiesファイルをロードする
  @Override
   public Navigation afterActionInvoke(ActionContext actionContext,
      MethodDesc targetMethod, Object page, Object[] args) {
     if (browserProperties.containsKey("X-UA-Compatible")) {
        String value = browserProperties.getProperty("X-UA-Compatible");
        actionContext.getResponse().setHeader("X-UA-Compatible", value);
     }
    return super.afterActionInvoke(actionContext, targetMethod, page, args);
  }
}
```

Seasar Conference
2009 Autumn



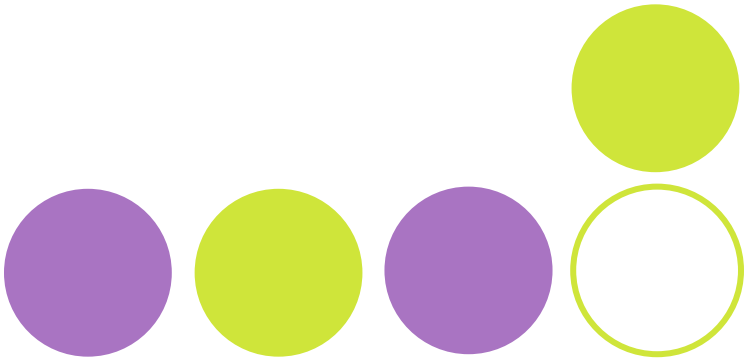


Exception Handler

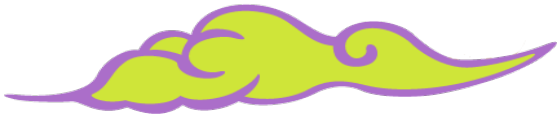


ExceptionHandler

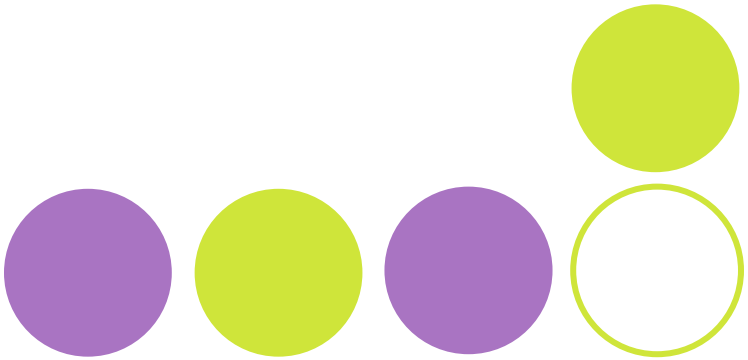
- 特定の例外を補足し処理をフックする
  - ExceptionHandler
    - 特定例外をキャッチ
  - GlobalExceptionHandler
    - 最後の砦。ServletExceptionをキャッチ
  - どちらもContainerAdapterのAPIから取得する。何を返すかはContainerAdapter次第。



Seasar Conference
2009 Autumn

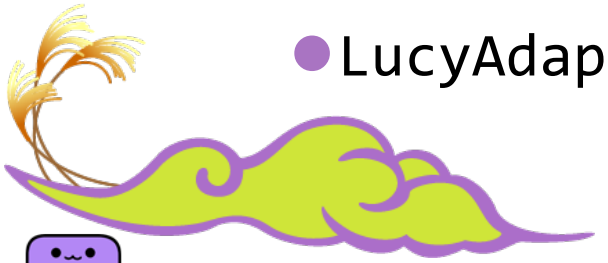


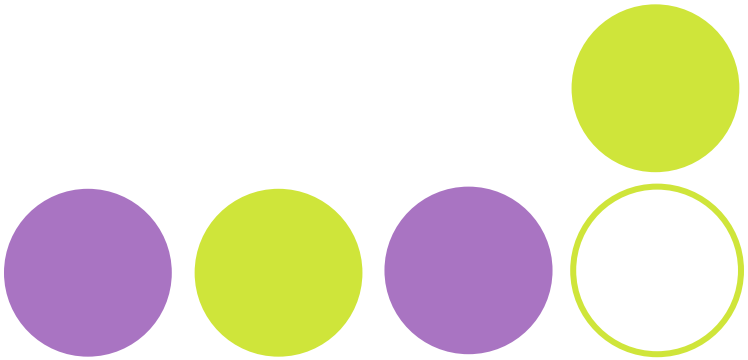
Container Adapter



ContainerAdapter

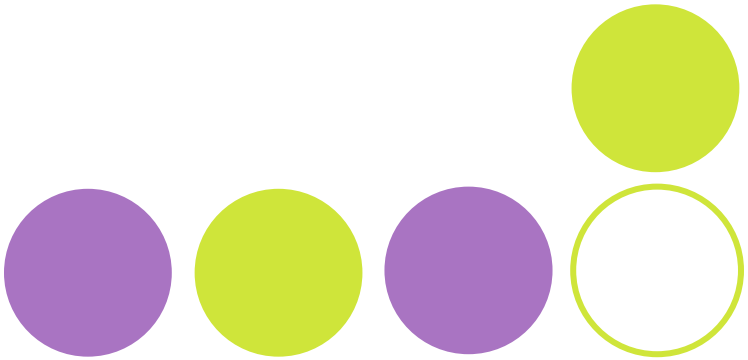
- DIコンテナとのアダプタ機構
  - デフォルトはSimpleContainerAdapter.
    - コンポーネントはPageのみでシングルトン固定
  - 大体のコンテナ用に用意した
    - T2-extモジュールのほうに入ってます。
    - Seasar2Adapter
    - SpringAdapter
    - GuiceAdapter
    - LucyAdapter





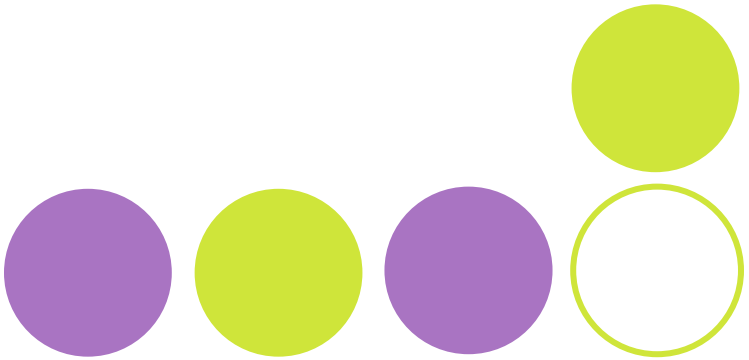
Seasar2Adapter

- Seasar2用のアダプタ
  - 実は2つある。
    - Seasar2Adapter
      - Seasar2.4用
    - Seasar2ClassicAdapter
      - Seasar2.3用
  - そこそこ頑張っている



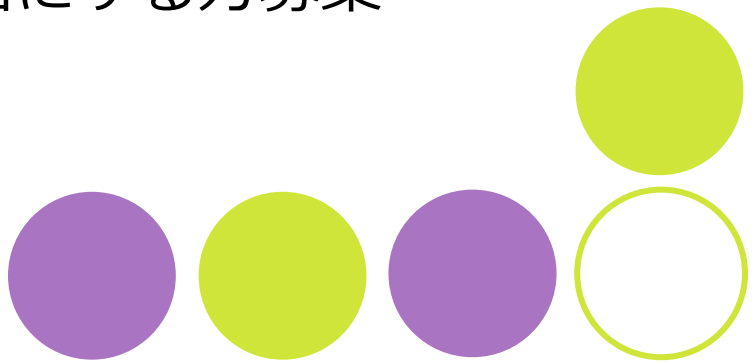
SpringAdapter

- Spring用のアダプタ
  - Spring2.5.x用に作成
  - 次バージョンで3.0系に対応する予定
  - 特に難しい事はしていないw



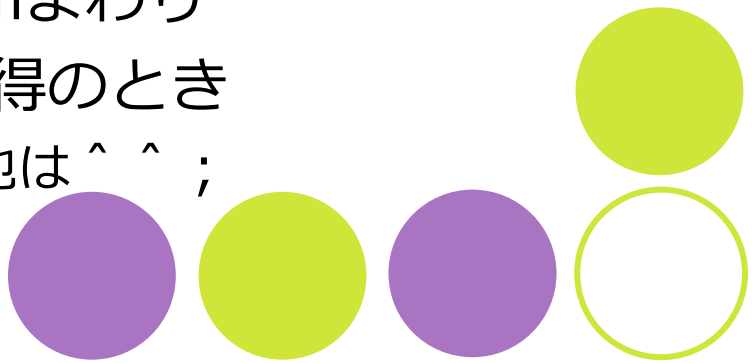
GuiceAdapter

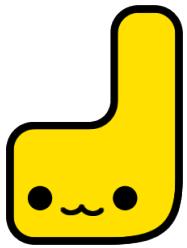
- Guice用のアダプタ
  - JDK6のServiceLoader経由でModuleの一覧をとってそれをInjectorに登録する
  - Guice1.0系用に作成。次バージョンでGuice2.0サポートの予定
  - 複数コンポーネント取得のところは今後工夫が必要。誰かフィードバックください♫
    - よねむら(yone098)と一緒にする方募集

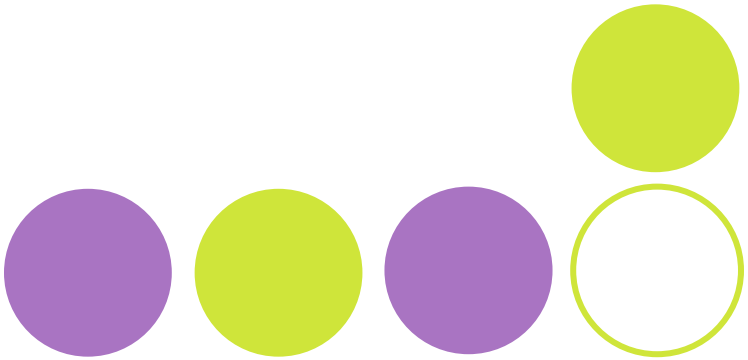


ContainerAdapter拡張

- 拡張するタイミング
  - 取得元が様々ある場合
    - Propertiesファイル
    - JNDI経由のLookup
    - 既存設定ファイルからパースする
  - 既存のContainerAdapterの実装ではうまくいかないとき
    - ExceptionHandler/Pluginまわり
    - 複数コンポーネント一括取得のとき
      - Seasar2/Lucyは大丈夫。他は＾＾；



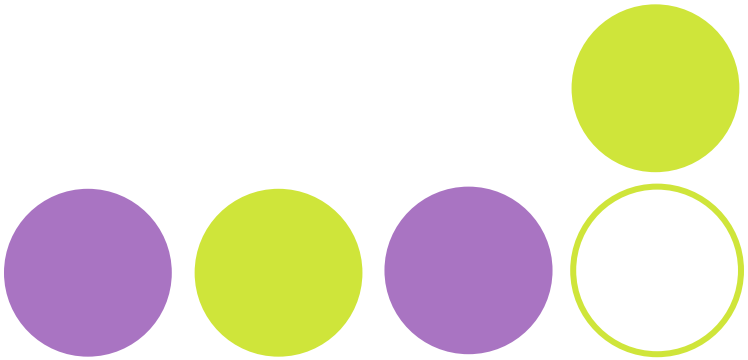
 まとめ



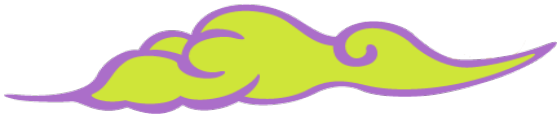
Seasar Conference 2009 Autumn

まとめ

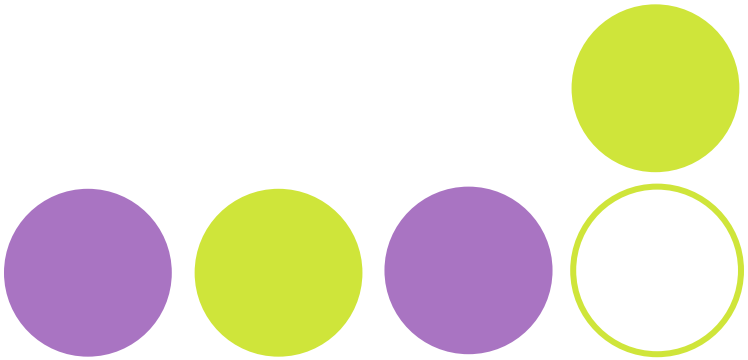
- T2は各コンポーネントが拡張可能
  - 大体ContainerAdapter経由
  - パフォーマンス向上や、不必要な機能をそぎ落として自分が使いやすい形にするのも一つ
    - 実際私はそうやって使っています
  - 自由に拡張してみてください
    - こうしたほうがいいよーとフィードバックくれると、ありがたいです。



Seasar Conference
2009 Autumn

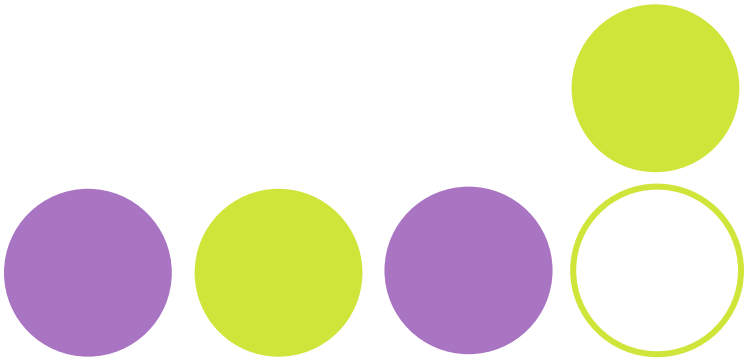


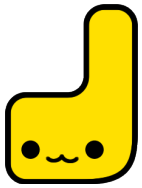
T2やるべ



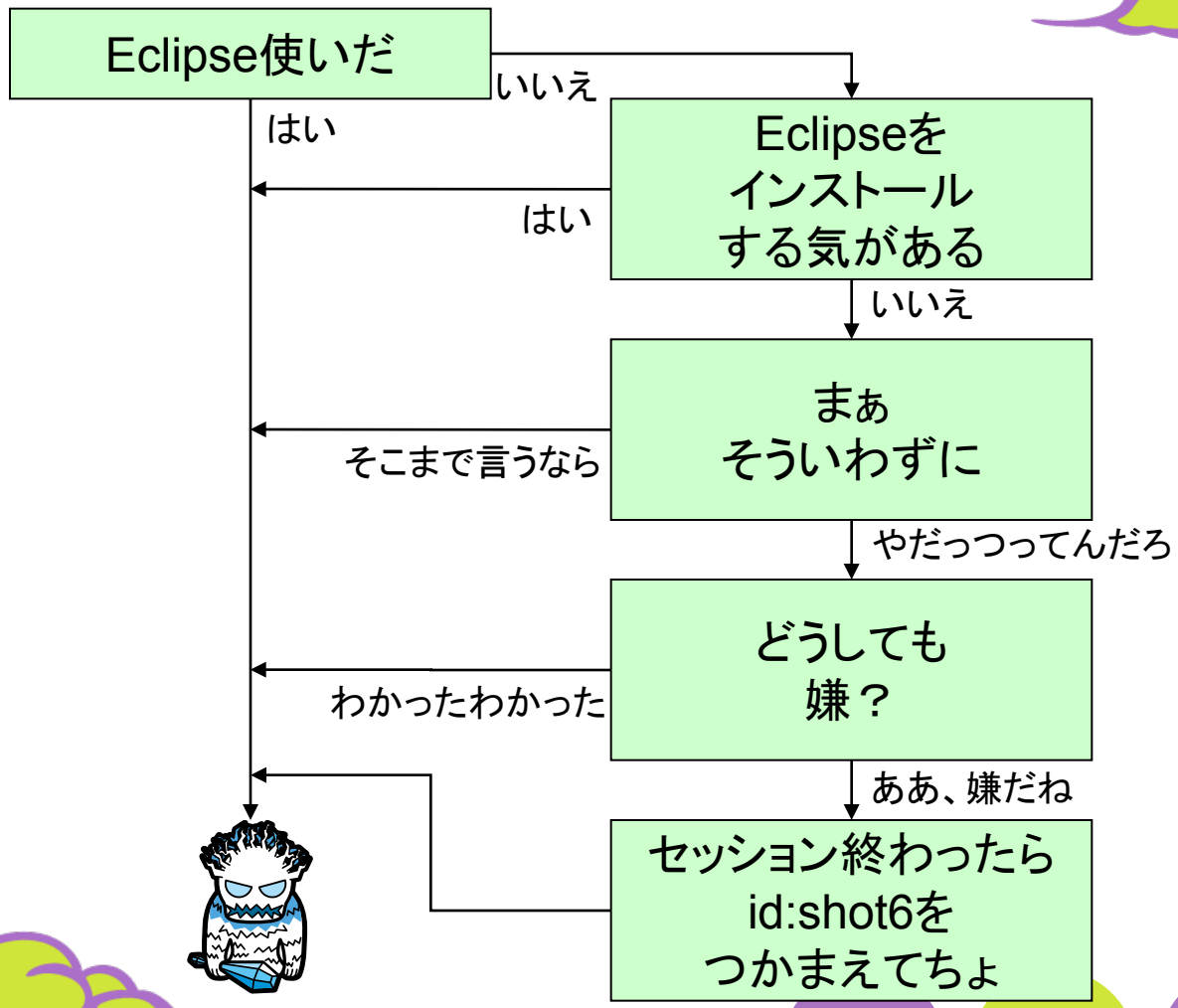
Vili

- T2をはじめたいけどどうすればいいの？





かんたん T2はじめよう チャート



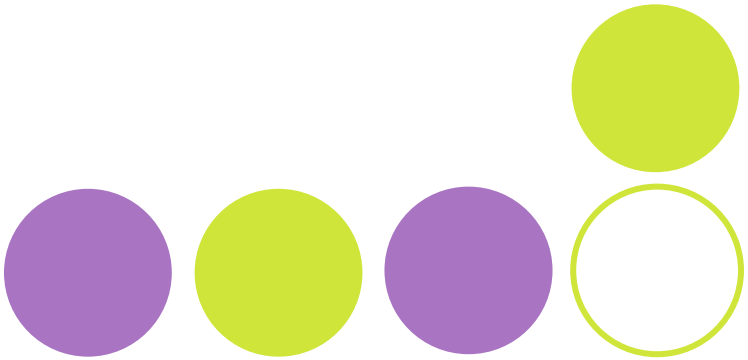
プロジェクト作成支援
Eclipseプラグイン
「Vili（ヴィリ）」
を使いましょう

Vili

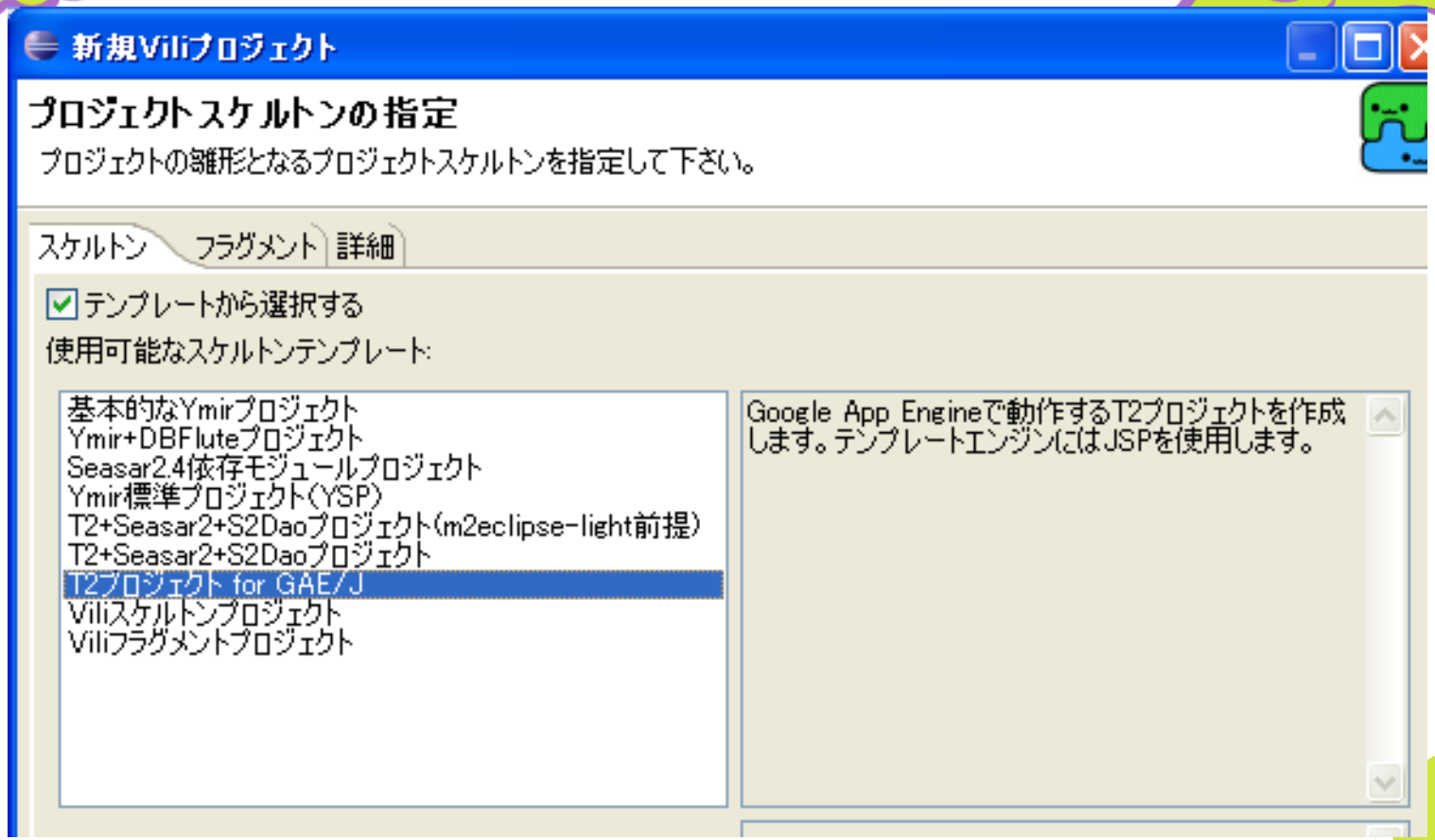
- Vili
  - 汎用のプロジェクト生成支援Eclipseプラグイン
  - http://code.google.com/p/t-2/wiki/Vili
  - 読み方は「ヴィリ」
    - 北欧神話の神様の名前。オーディンの兄弟
  - プロジェクトの雛形（スケルトン）を用意することで様々なプロジェクトを生成可能
    - T2プロジェクト、Ymirプロジェクト、Cubbyプロジェクト、…
  - プログラム部品（フラグメント）を用意することでプロジェクトに簡単に機能を追加可能
    - データベースアクセス機能、メール送信機能、統合テスト環境、ログイン機能、…

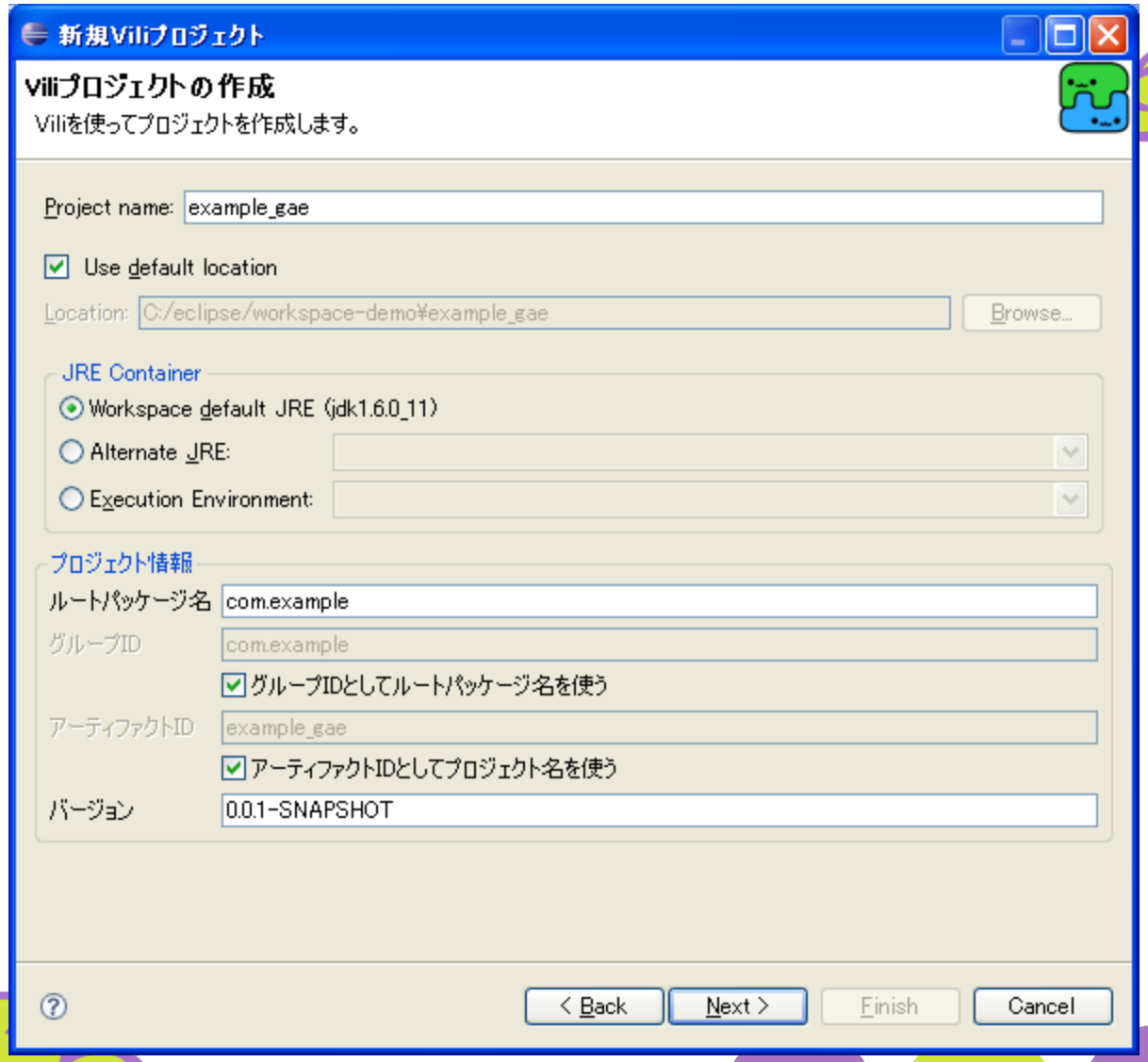
Vili

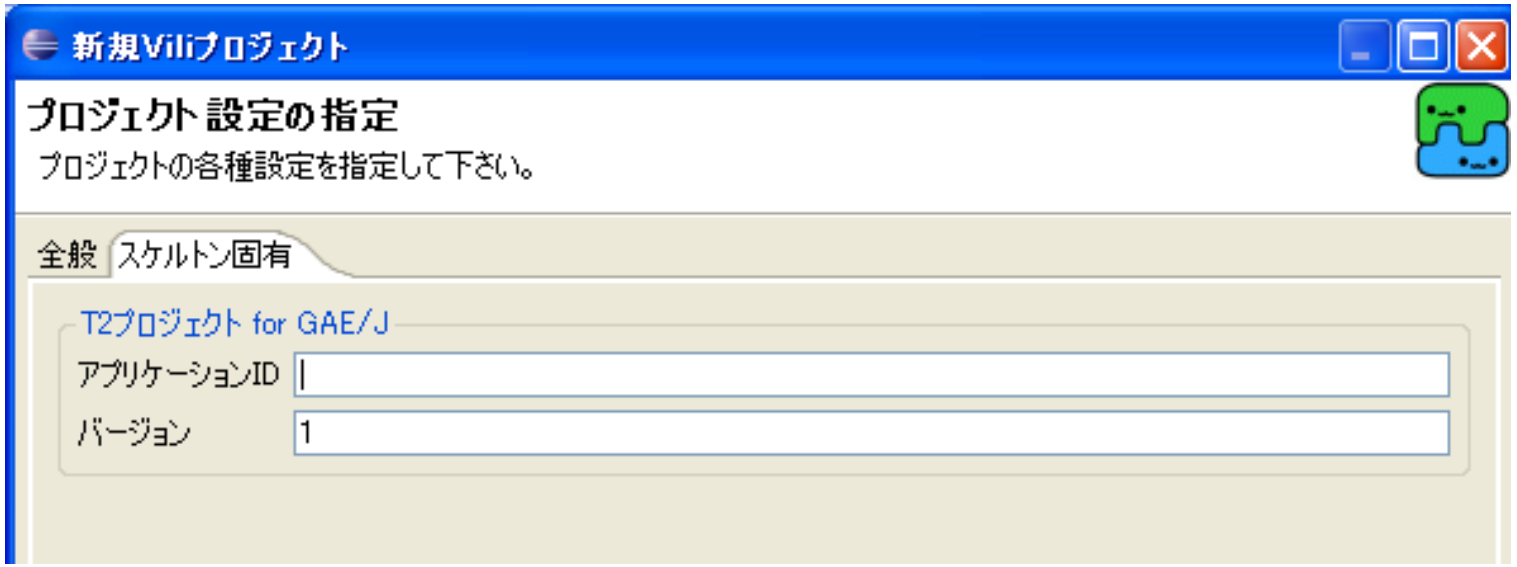
- T2用のスケルトン
  - T2+Seasar2+S2Daoプロジェクト
  - T2プロジェクト for GAE/J
- T2用のフラグメント
  - 今はまだありません
- スケルトン、フラグメントは今後増やしていく予定です

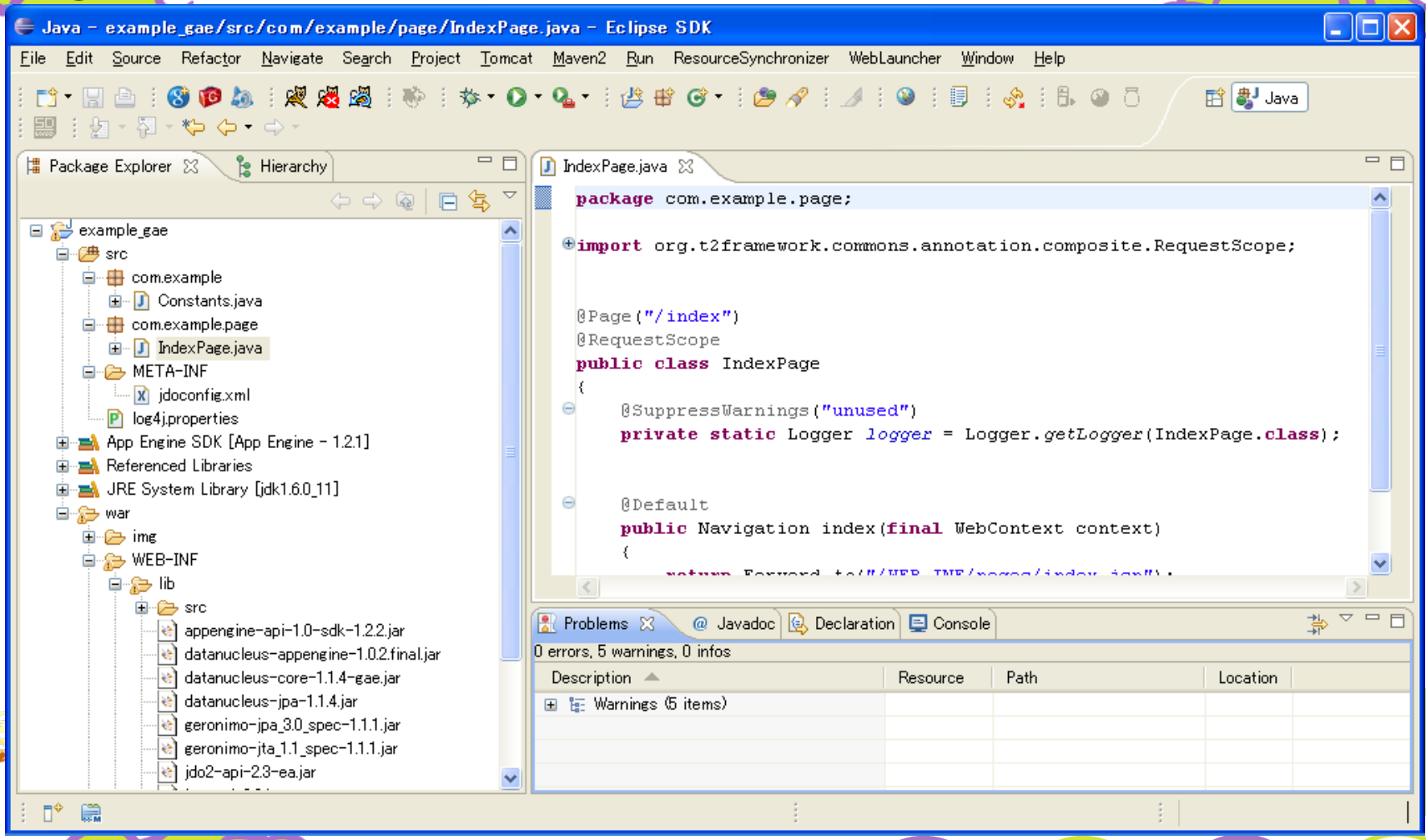


使い方





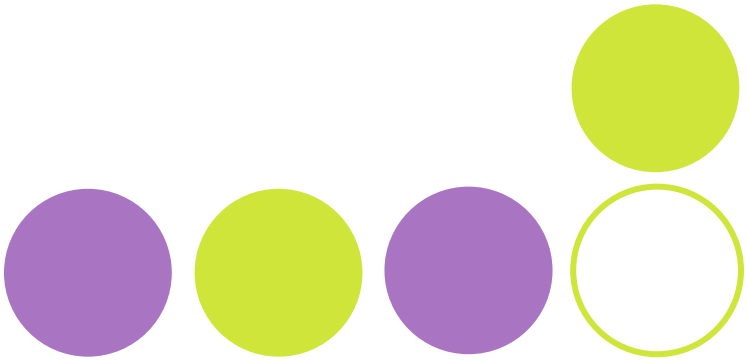


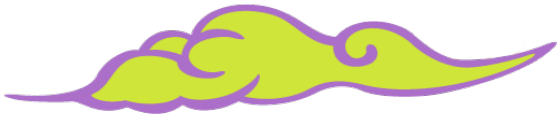


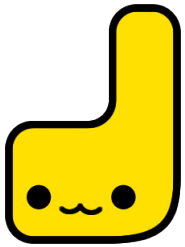

まとめ

Vili

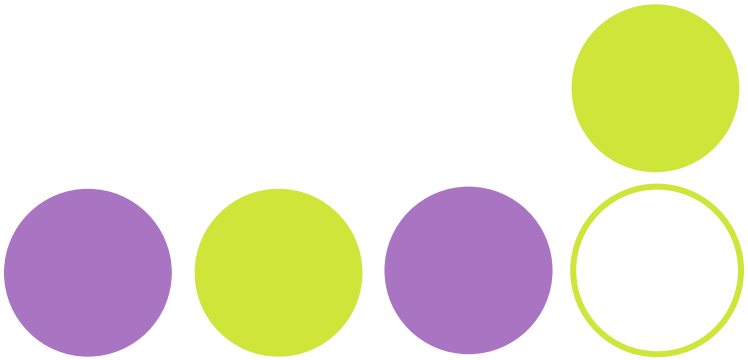
- ドキュメントがほとんどないので今後公開していく予定です
- T2用のスケルトン、フラグメントも今後増やしていく予定です





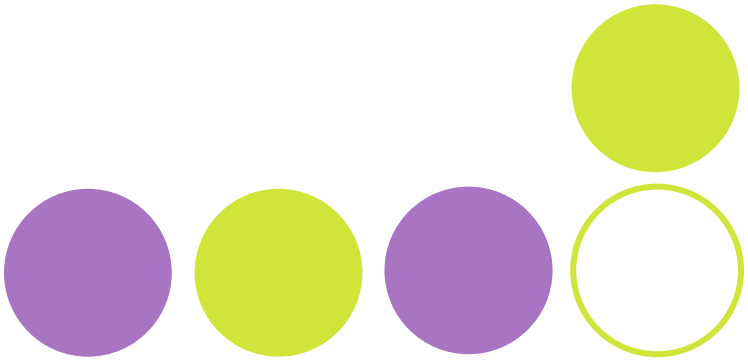


AMFで本気出す



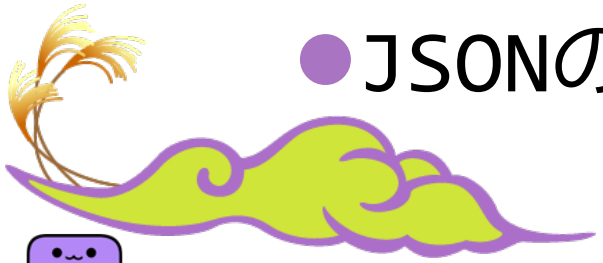
T2 ～AMF～

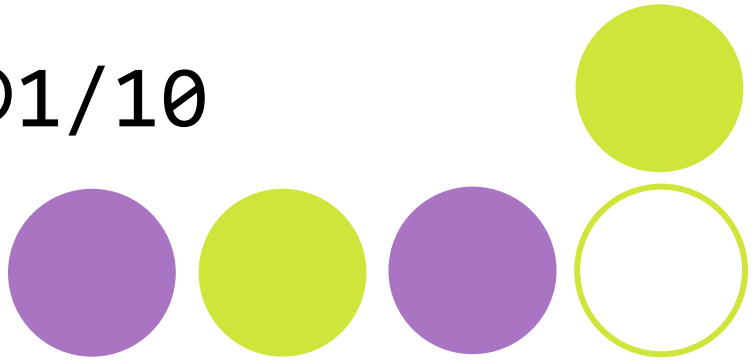
- そもそもAMFとは？



T2 ～AMF～

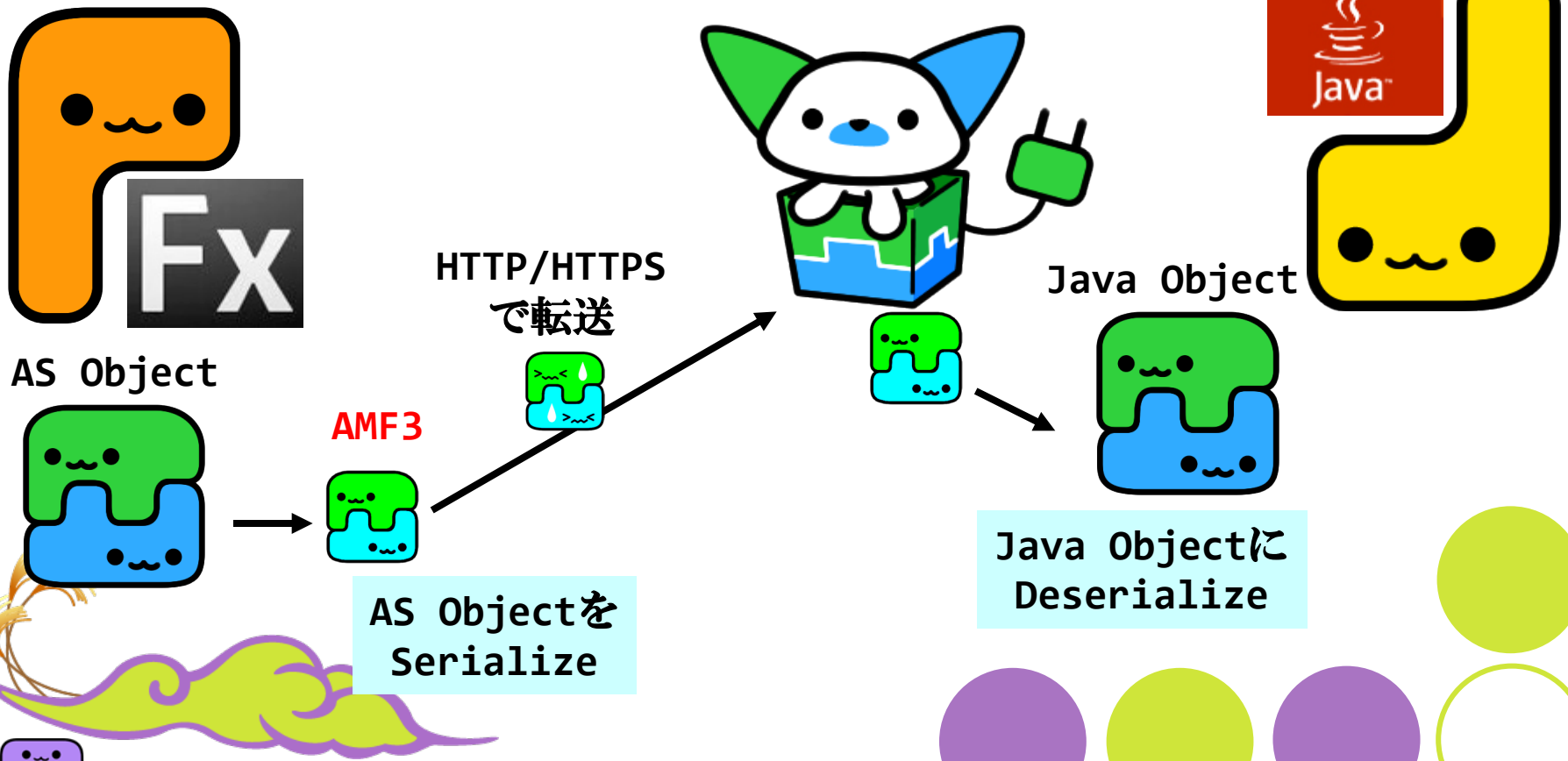
- AMF
  - ActionMessageFormat
  - Flash/FlexのObject Serialize方法（オブジェクト<->AMFデータ）
  - 通信フォーマットに利用可能
  - バイナリ形式で小さい！
    - JSONの1/4、XMLの1/10



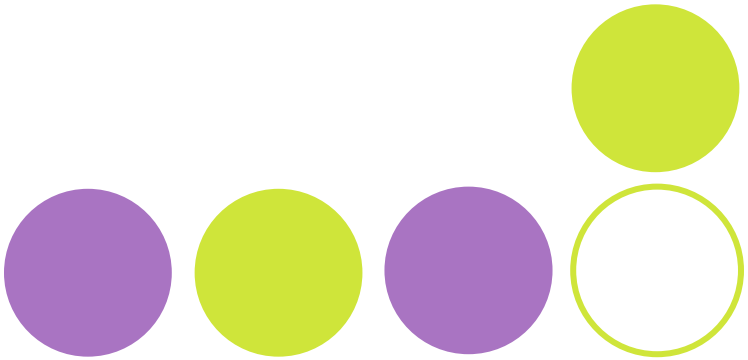


T2 ～AMF～

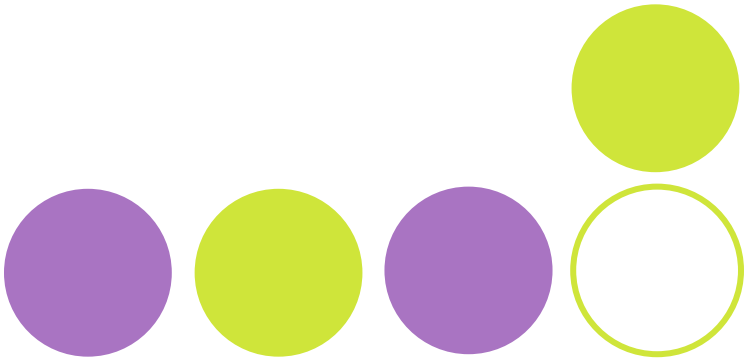
●概要図



AMF通信可能イコール Flash/Flexと やりたい放題

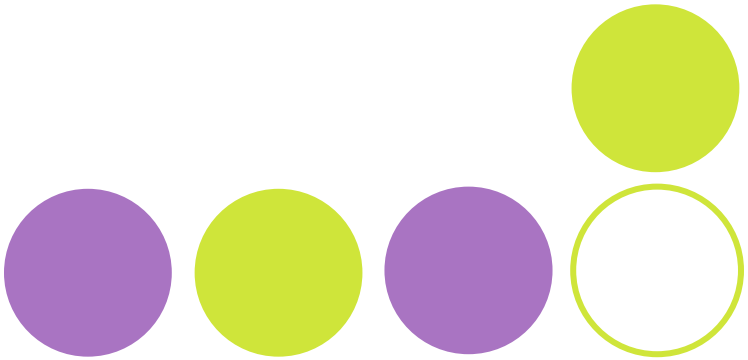


- T2AMF詳細



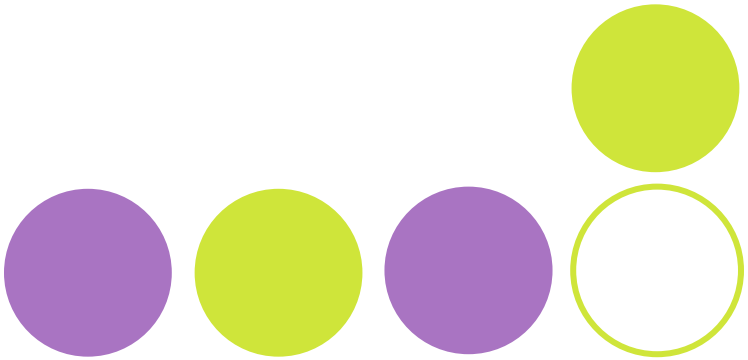
T2 ～AMF～

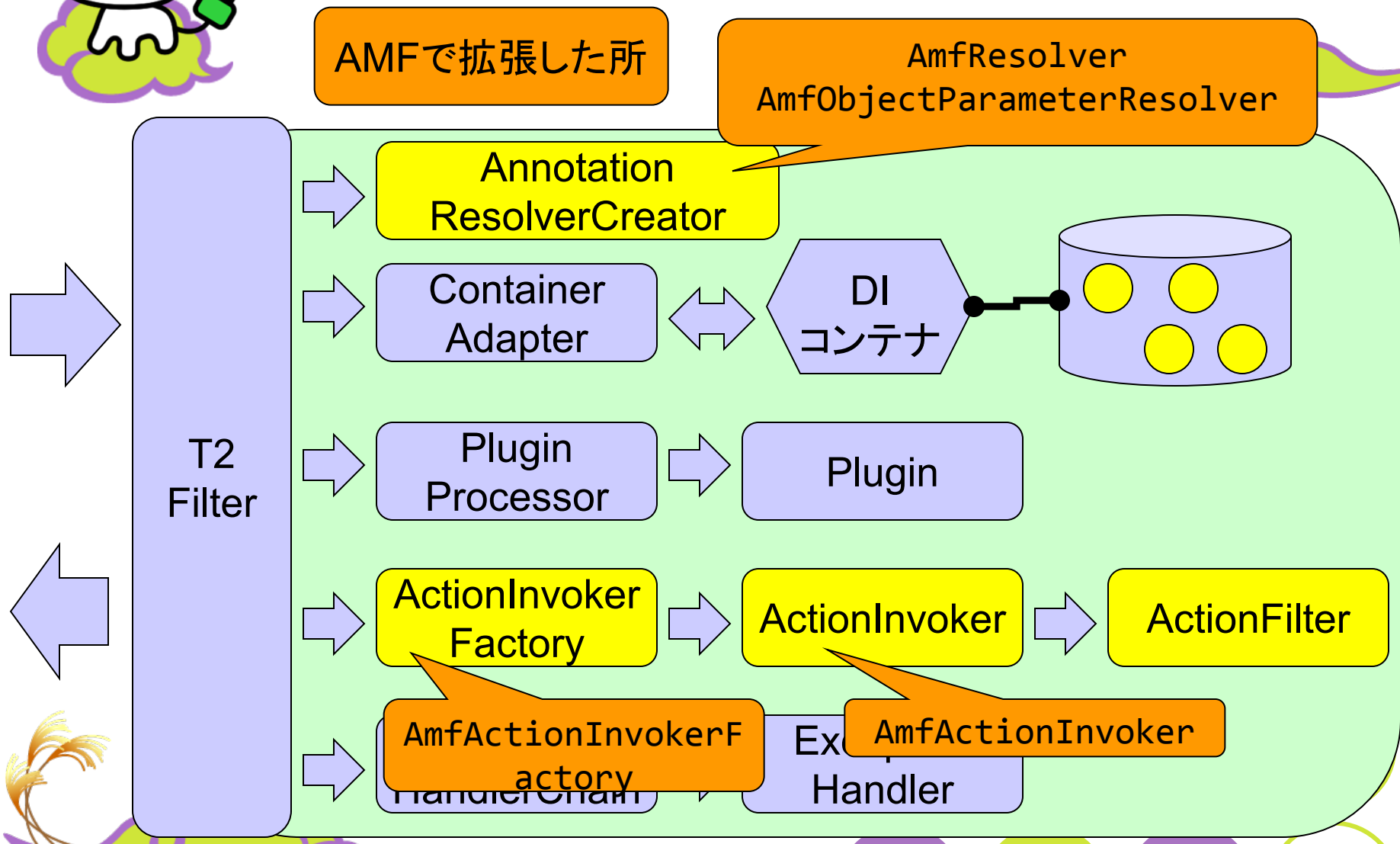
- T2の標準的な方法で拡張されてます！
- AmfResolver
  - @Amfを利用可能にする
- AmfObjectParameterResolver
  - アクションメソッドの引数を解決



T2 ～AMF～

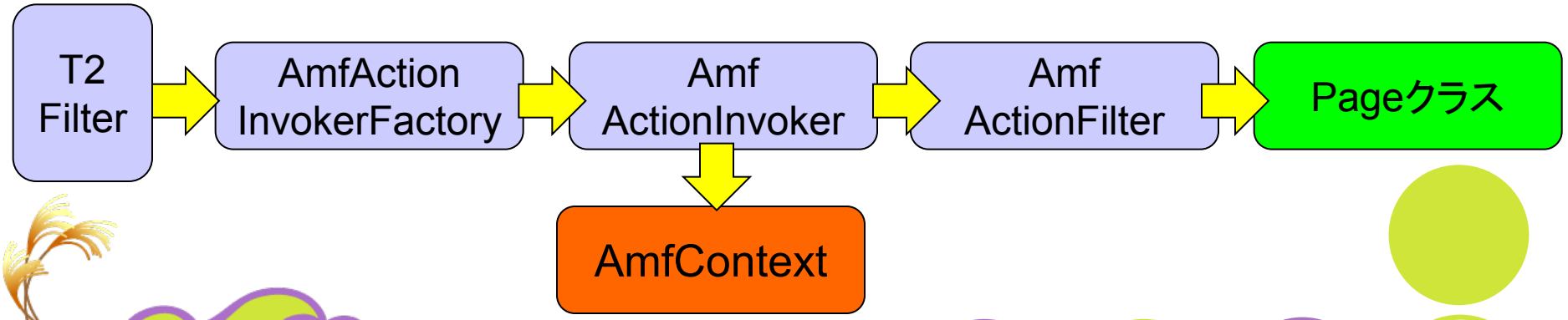
- AmfActionInvokerFactory
  - AmfActionInvokerを生成
  - T2がAMFだけしか扱わなくできます
- AmfActionInvoker
  - AmfContext（後述）を生成・利用します
  - AMF用のActionFilterを生成・呼び出しします





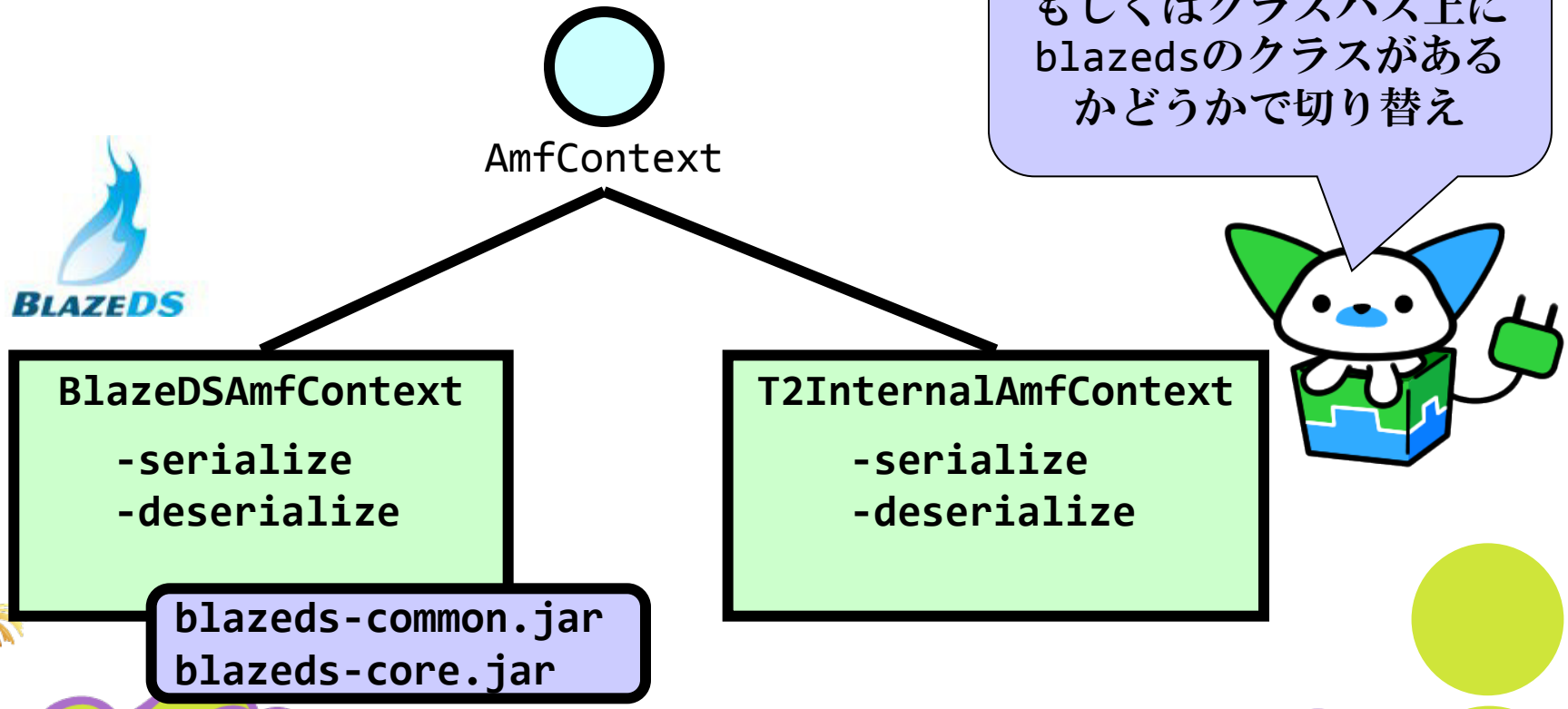
T2 ～AMF～

- AmfContext
  - AMFのシリアライズ・デシリアライズ担当
  - AMFに関する操作はすべてこのインターフェース経由
  - こいつが肝！



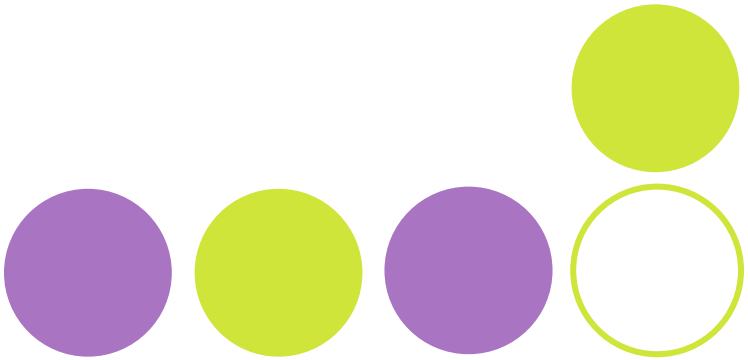
T2 ～AMF～

- AmfContextの実装は２つ

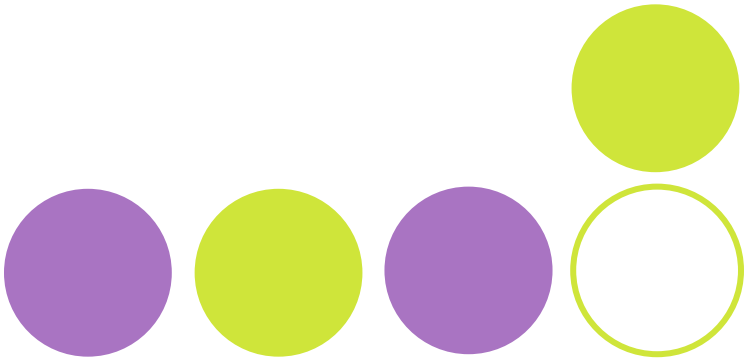


T2 ～AMF～

- T2InternalAmfContext
  - S2Flex2をベースにしたAMF実装
    - すばらしいAMF実装（コミッターの方に感謝！）
    - 依存ライブラリなしでAMFの利用が可能
- BlazeDSAmfContext
  - BlazeDSに委譲（丸投げ）
    - blazeds-common.jarとblazeds-core.jarが必要

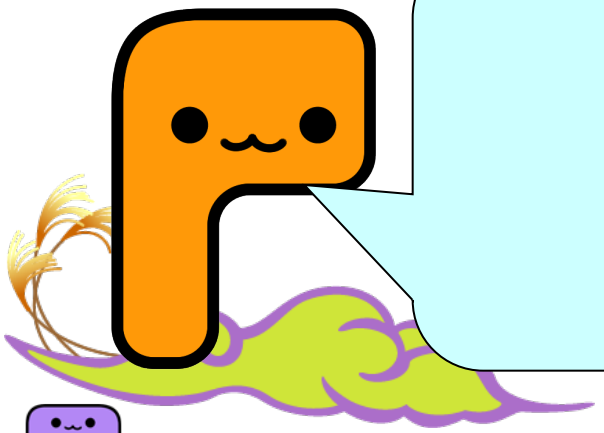


●使い方



T2 ～AMF～

- 使い方
  - まずは「destination」を決めよう
    - destination=呼び出すPageクラスを示すあて先
  - つぎに呼び出すメソッドを決めよう
    - 引数と戻り値も決めよう



あて先は「unicef」
メソッドは「bokin」
引数はint
戻り値はString
でいこう！

T2 ～AMF～

- 決めたdistinationとメソッドでサーバ側を作成

```
@Page("unicef")  //destination
public class UnicefPage {

    @Amf  //メソッドにつける @Amf("bokin")でも可
    public Navigation bokin(int bokingaku){

        Unicef.sendToAfrica(bokingaku);

        //AmfResponseに戻り値を入れる
        return AmfResponse.to("Thank you!");
    }
}
```

T2 ～AMF～

- クライアント側
  - まずは結果を受け取るハンドラを作成

```
//通信成功時に呼ばれる
public function handleResult(e:ResultEvent,
                             token:Object=null):void{
    var message:String = String(e.result);
    Alert.show(message);//”Thank you”と表示
}
//通信失敗時に呼ばれる
public function handleFault(e:FaultEvent,token:Object=null):void{
    log.error(ObjectUtil.toString(e));
}
```

- 最後に呼び出し部分

```
//呼び出し用RemoteObject作成
var unicefPage:RemoteObject = new RemoteObject("unicef");
//接続設定（お決まり）
var endPoint:String =
          URLUtil.getFullURL(Application.application.url,“t2.amf");
var channel:Channel = URLUtil.isHttpsURL(endPoint)?
        new SecureAMFChannel(null,endPoint):
        new AMFChannel(null,endPoint);
var channelSet:ChannelSet = new ChannelSet();
channelSet.addChannel(channel);
unicefPage.channelSet = channelSet;
//サーバ呼び出し
var token:AsyncToken = unicefPage.bokin(1000000);
token.addResponder(
          new AsyncResponder(handleResult,handleFault));
```

T2 ～AMF～

- もし送受信にデータクラスを使う場合は

```
//aliasで、このデータクラスのデータを入れるJava側のクラスを指定
[RemoteClass(alias="unicef.java.UnicefDto")]
public class UniceDto
{
  public var bokingaku:int;
  public var message:String;
}
```

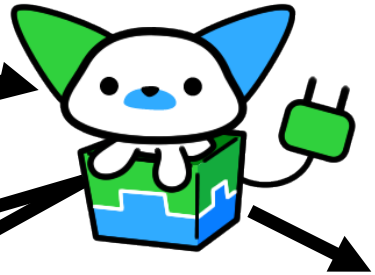
- Java側に、クライアントのデータクラスと同じ構造のクラスを用意し、メタデータで指定。

T2 ～AMF～

- まとめると

RemoteObject作成
unicefPage.bokin(10000)

destination:unicef
メソッド：bokin
引数：1



@Page(“unicef”)
UnicefPage
@Amf
bokin(int money)

呼び出し成功：
ResultEvent
result:”Thank you!”

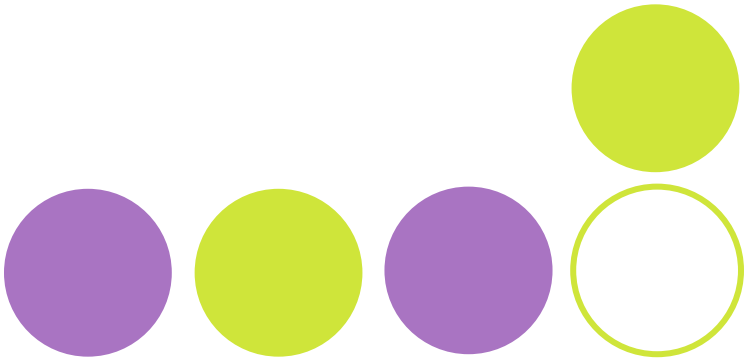
handleResult
メソッド

呼び出し失敗：
FaultEvent
message:エラー内容

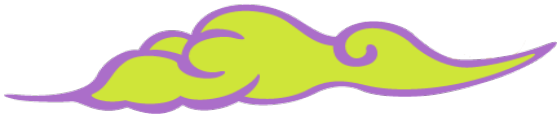
handleFault
メソッド

T2 ～AMF～

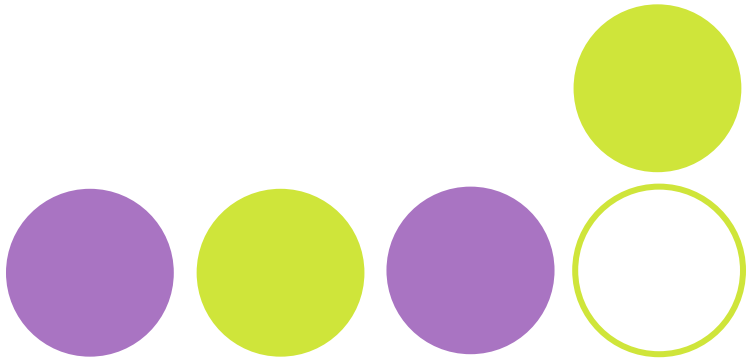
- demo



Seasar Conference
2009 Autumn

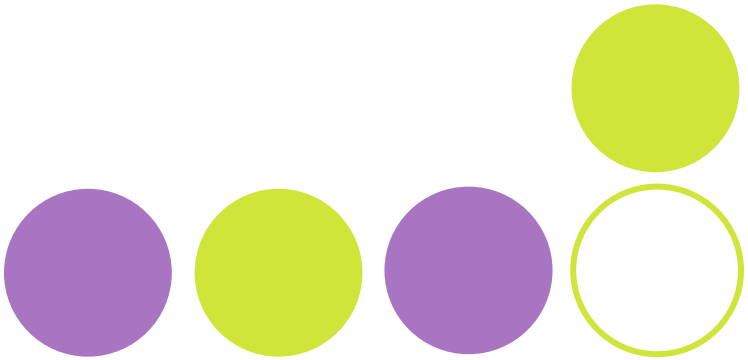


T2 Hack



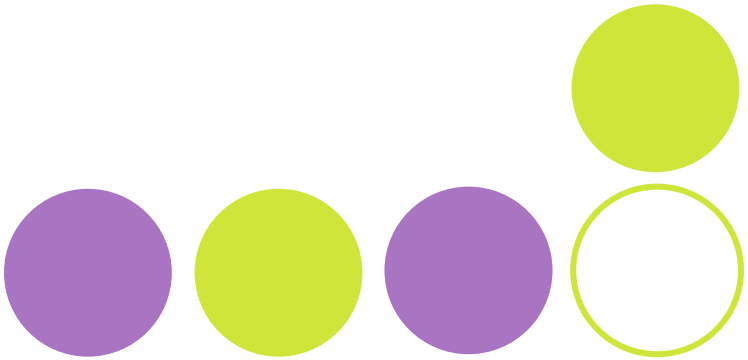
T2 Hack Agenda

- T2とScala連携
- T2とGroovy連携
- T2とJavaFX連携
  - AMF通信
- T2とApplet連携



T2 Hack with Scala

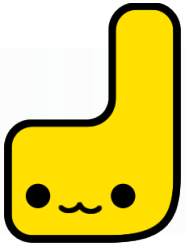
- Scala + T2



Seasar Conference 2009 Autumn

T2 Hack with Scala

- Eclipseでの開発
  - Scala pluginをinstall
    - http://www.scala-lang.org/scala-eclipse-plugin-nightly
  - Project設定
    - Add Scala Nature
  - すぐ試したい人は
    - t2-samples-scalaをdownload



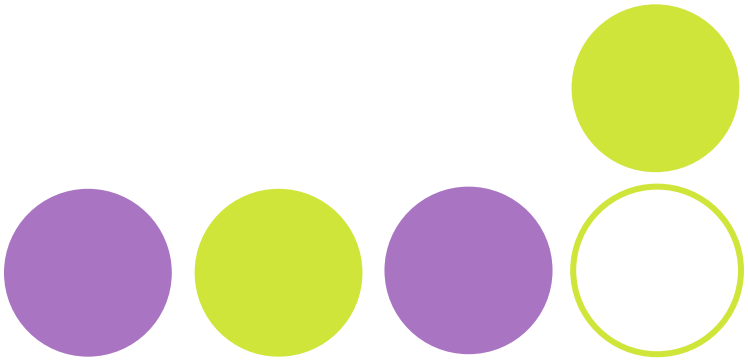
T2 Hack with Scala

- Scala Page class

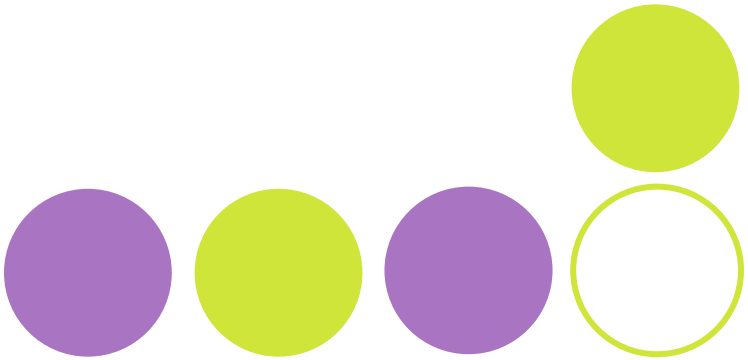
```
package examples.page {
  import org.t2framework.t2.navigation._
  @Page("/scala" )
  class ScalaPage {
    @Default
    def index(w: WebContext): Navigation {
      println("=== index ===");
      Redirect.to("/jsp/scala.jsp")
    }
    @POST
    @ActionParam
    def test(w: WebContext): Navigation {
      println("=== index ===")
      Forward.to("/jsp/scala.jsp")
    }
  }
```

T2 Hack with Scala

- Demo



Groovy + T2



T2 Hack with Groovy

- Eclipseでの開発
  - Groovy pluginをinstall
    - http://dist.codehaus.org/groovy/distributions/update/
  - Project設定
    - Add Groovy Nature
    - Groovy Properties => output location変更
    - 存在しないフォルダを指定すると嵌る
  - すぐ試したい人は
    - t2-samples-groovyをdownload

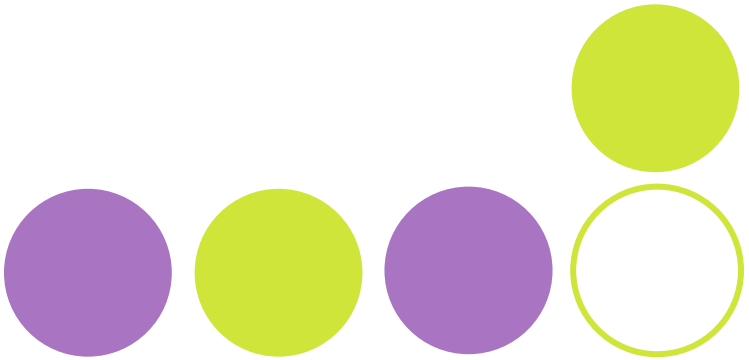
T2 Hack with Groovy

- Groovy Page class

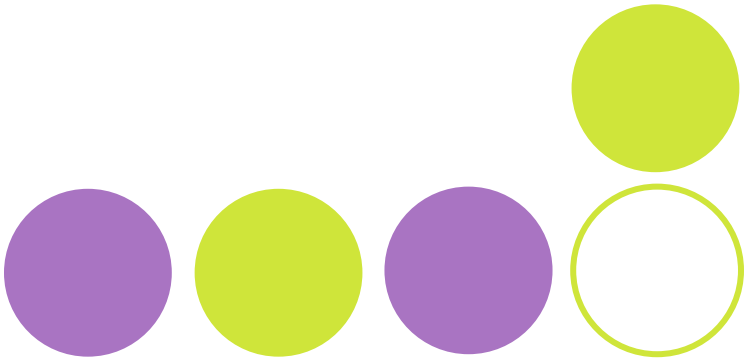
```
package examples.page
import org.t2framework.t2.annotation.core.Page
import org.t2framework.t2.spi.Navigation
import org.t2framework.t2.contexts.Request
import org.t2framework.t2.navigation.Forward
import org.t2framework.t2.annotation.core.Default

@Page(“/groovy” )
public class GroovyPage {
  @Default
  def Navigation index(Request req) {
    req.setAttribute(“message”, “Hello Groovy”);
    println(“=== index Groovy Page ===“);
    Forward.to(“/jsp/groovy.jsp”)
  }
}
```

T2 Hack with Groovy

- demo



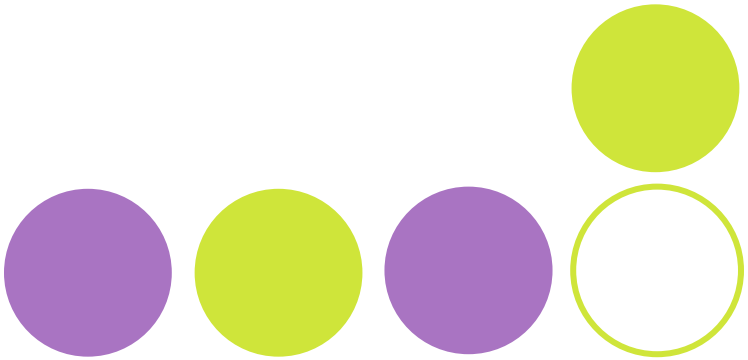
JavaFX + T2



Seasar Conference 2009 Autumn

T2 Hack with JavaFX

- Eclipseでの開発
  - しない＞＜
- NetBeansで開発(ver 6.7.1)
  - JavaFX Script pluginをinstall
  - T2 AMFと通信
    - Libraryを追加（t2-amf.jar）を追加
    - sandbox, jar1つでAMF可能



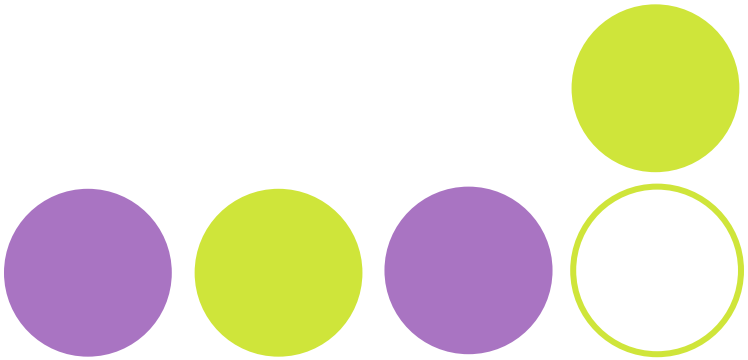
T2 Hack with JavaFX

- JavaFX sample code

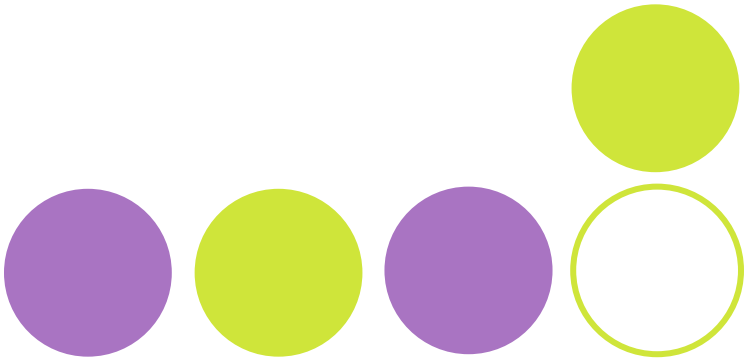
```
import org.t2framework.t2.format.amf.client.AmfConnection;

var slider:SwingSlider = SwingSlider {
  onMouseDragged: function(e) {

    var t2amf : AmfConnection = new AmfConnection();
    t2amf.setDestination("javapiano");
    t2amf.setOperation("play");
    t2amf.connect("http://localhost:8080/t2-haiku/t2.amf");
    t2amf.callAmf(e.dragX);

  }
}
```

T2 Hack with JavaFX

- demo

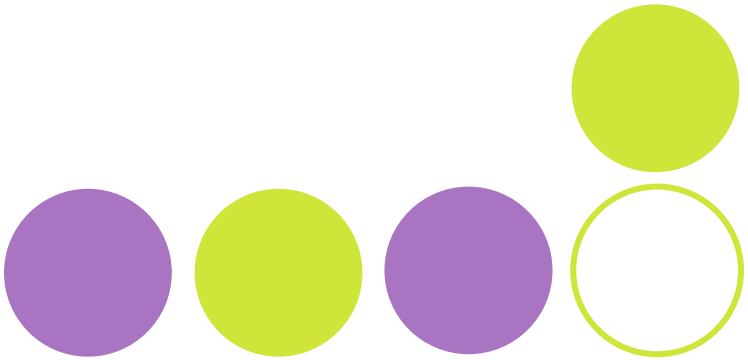


Applet + T2



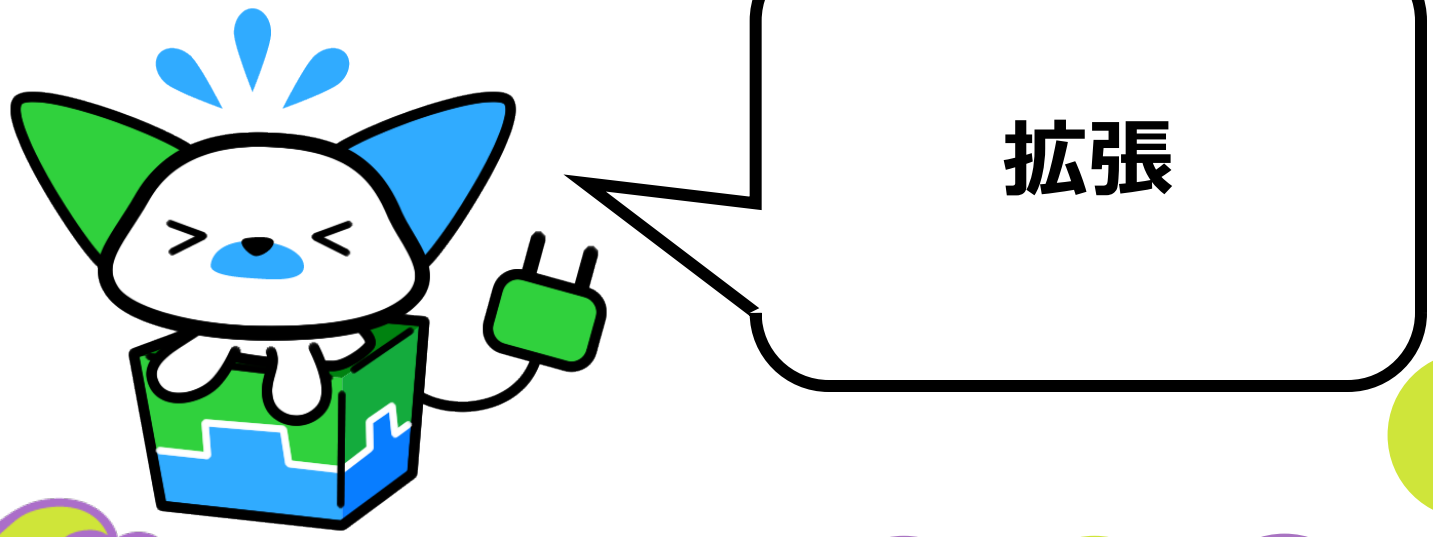
T2 Hack with Applet

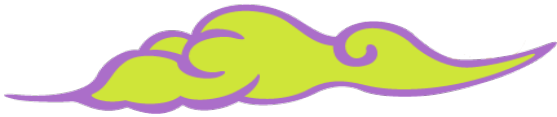
- demo



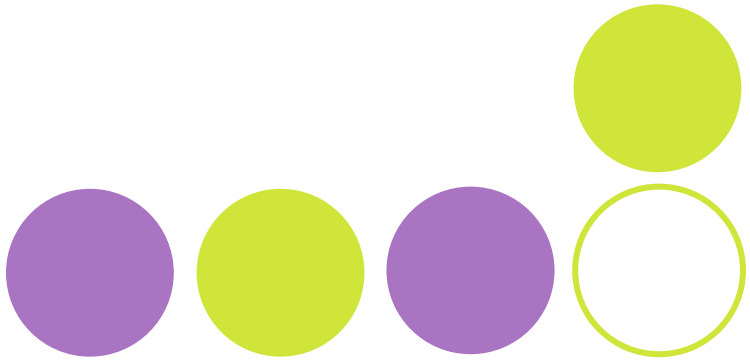
まとめ

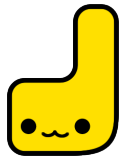
- T2拡張は簡単
- あなたのやる気次第



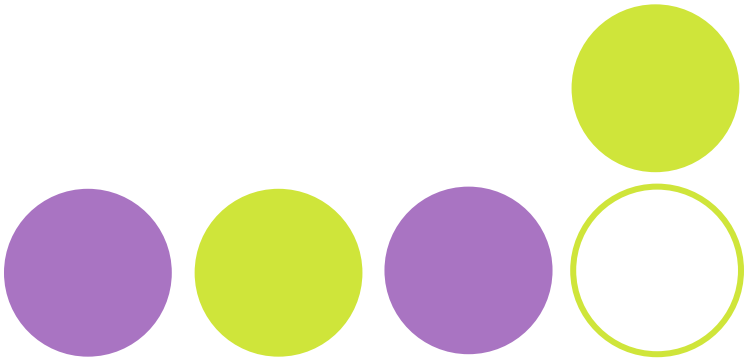


Enjoy T2!





一句



一句

雨の日に
お越し頂き
いとうれし♥

Seasar Conference
2009 Autumn

謝辞

御清聴
ありがとう
ございました