革新を続けるZopeの特徴と現在

Webアプリケーションの最先端基盤

オープンソースカンファレンス 2009 Kochi
2009年11月14日（土曜日）

山本 烈 (iccm inc.)
日本Zopeユーザー会(JZUG)

Zopeとは

複雑なWebアプリケーションを

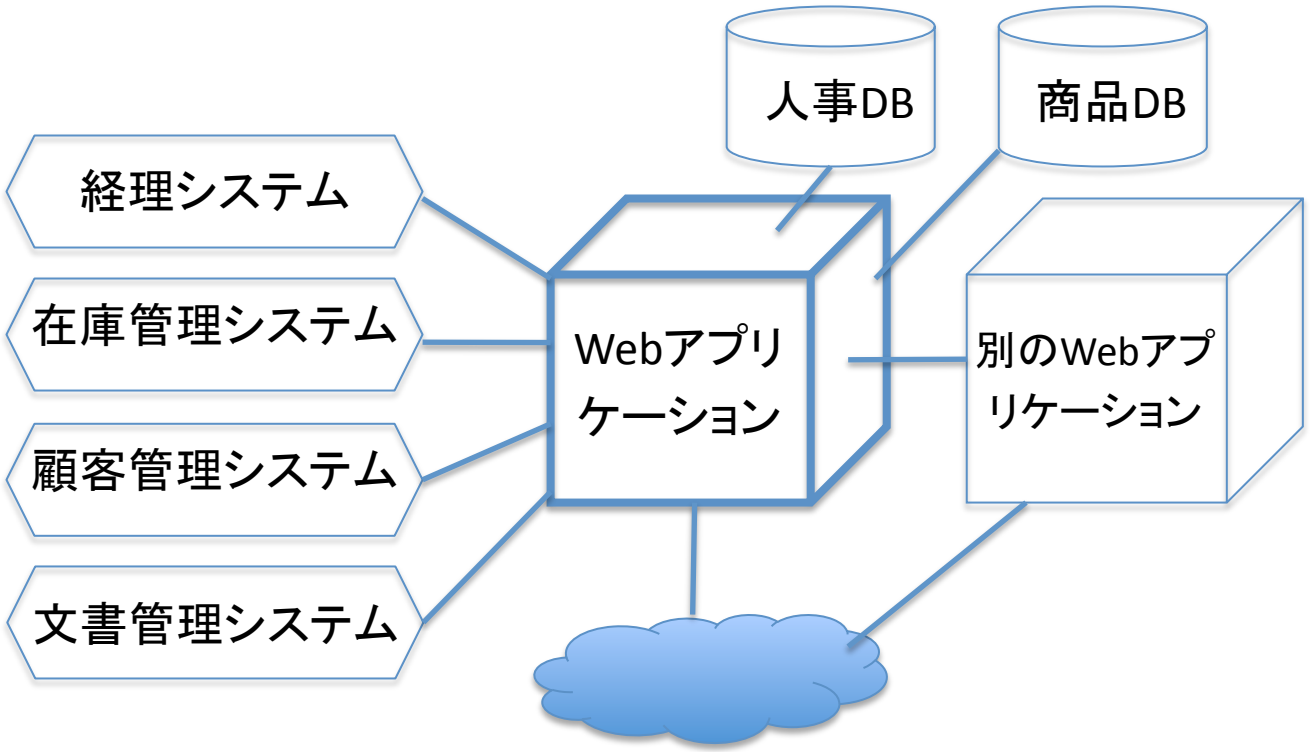
短期間で

低コストで

開発するソフトウェアツール

Webアプリケーションって？

複雑で大規模なアプリケーションをWeb技術を使って実現したもの

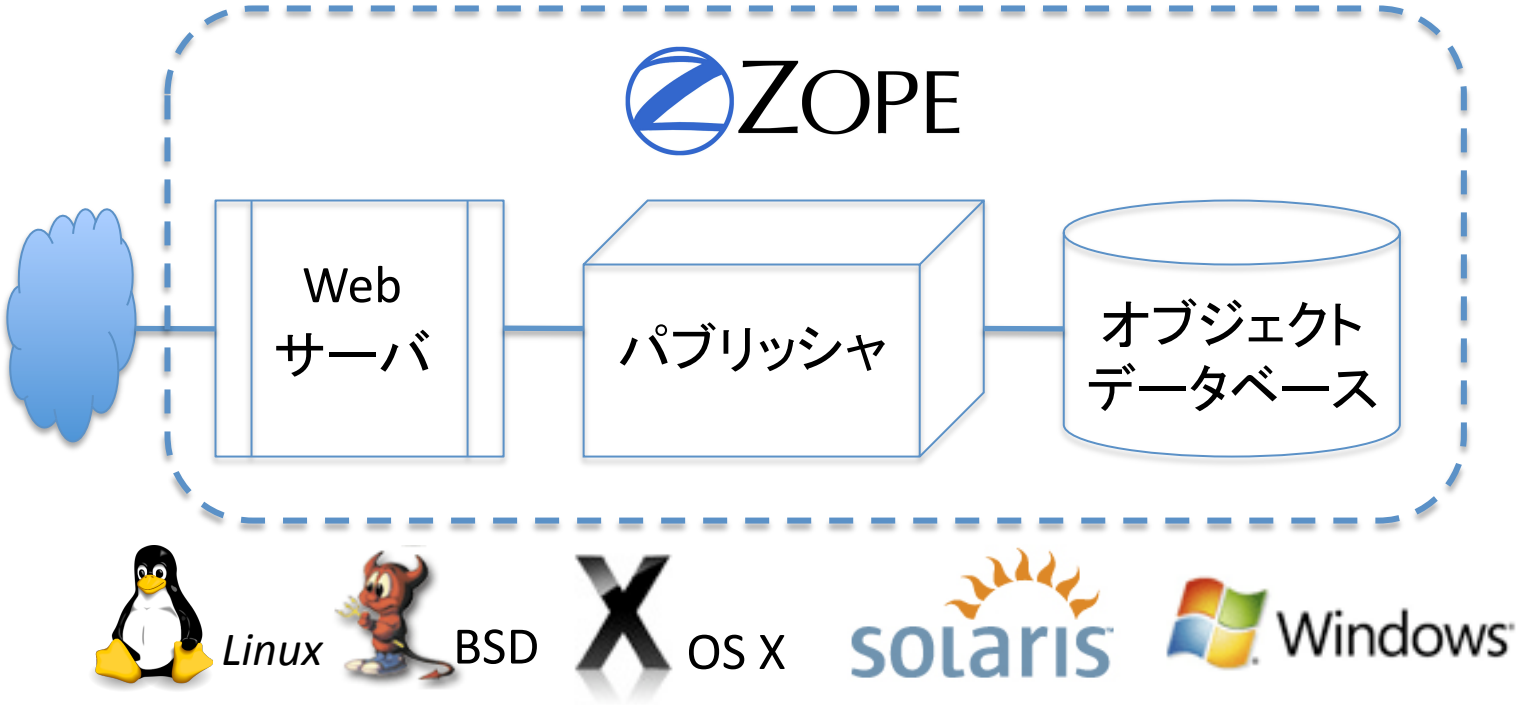


Zope

Z Object Publishing Environment

Webアプリケーション環境

1998年にオープンソースとして公開

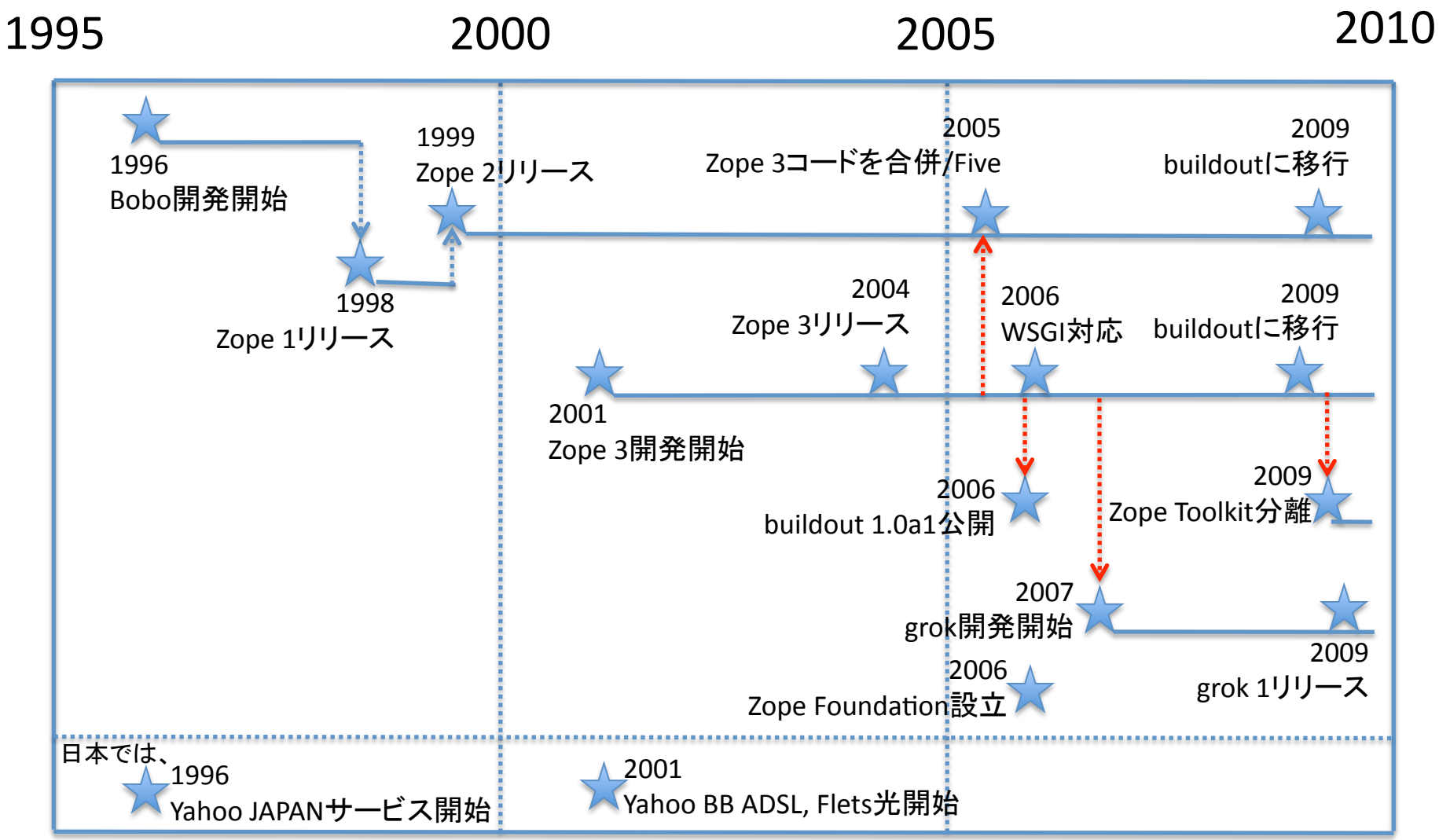


Zopeの特徴

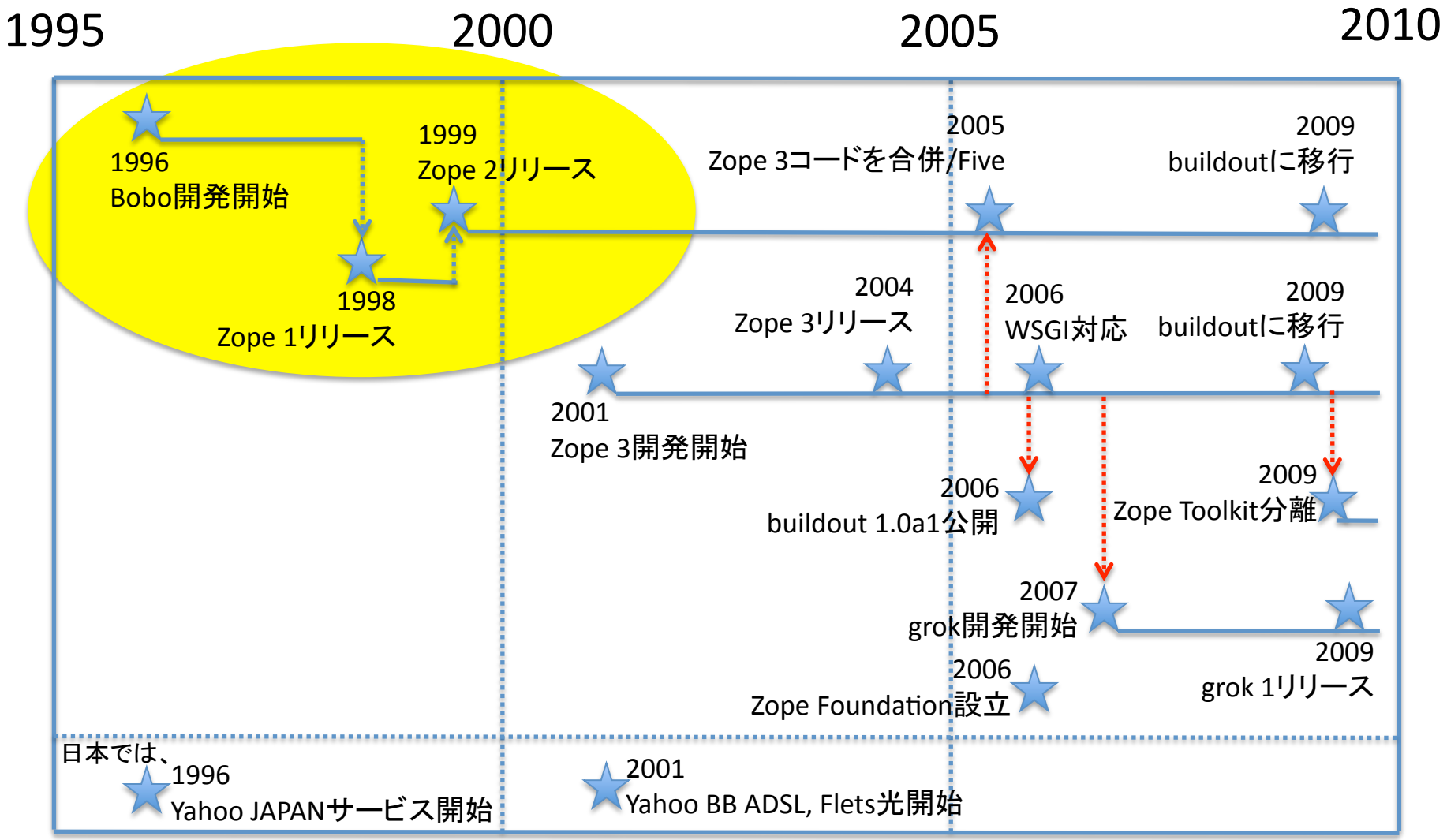
- 愚直なまでに目的に忠実
- 前衛的な革新を続ける
- そのために、なかなか理解されない

- 何が凄いのかを見ていきます

Zope年表


1995
2000
2005
2010
1996
Bobo開発開始
1998
Zope 1リリース
1999
Zope 2リリース
2005
Zope 3コードを合併/Five
2009
buildoutに移行
2001
Zope 3開発開始
2004
Zope 3リリース
2006
WSGI対応
2009
buildoutに移行
2006
buildout 1.0a1公開
2009
Zope Toolkit分離
2007
grok開発開始
2009
grok 1リリース
2006
Zope Foundation設立
日本では、
1996
Yahoo JAPANサービス開始
2001
Yahoo BB ADSL, Flets光開始

最初の波

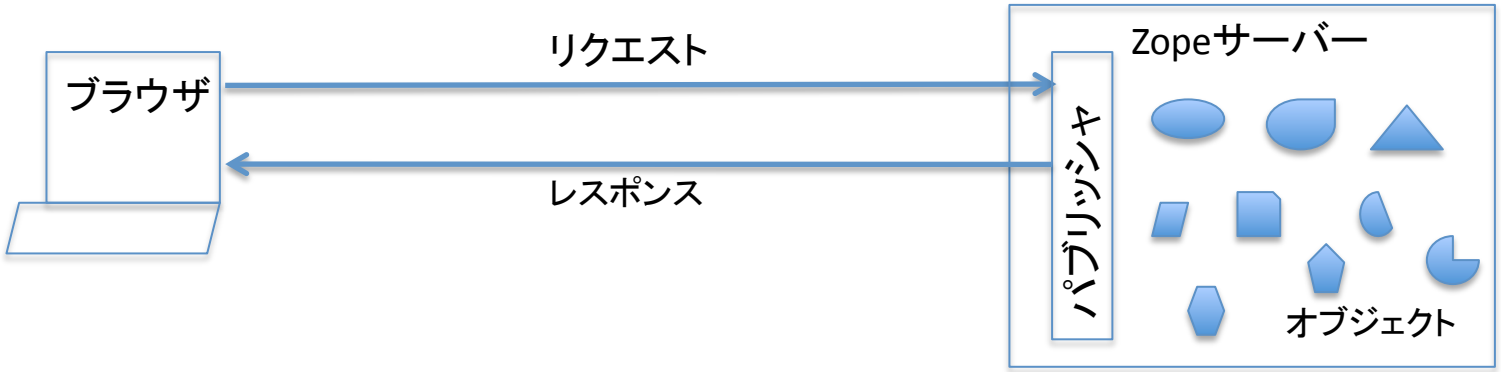

1995
2000
2005
2010
1996
Bobo開発開始
1998
Zope 1リリース
1999
Zope 2リリース
2005
Zope 3コードを合併/Five
2009
buildoutに移行
2004
Zope 3リリース
2006
WSGI対応
2009
buildoutに移行
2001
Zope 3開発開始
2006
buildout 1.0a1公開
2009
Zope Toolkit分離
2007
grok開発開始
2009
grok 1リリース
2006
Zope Foundation設立
日本では、
1996
Yahoo JAPANサービス開始
2001
Yahoo BB ADSL, Flets光開始

Zopeの革新

1998年 Zope登場

- オブジェクトパブリッシング
- トラバーシング
- オブジェクトデータベース
- オールインワン
- オープンソース化

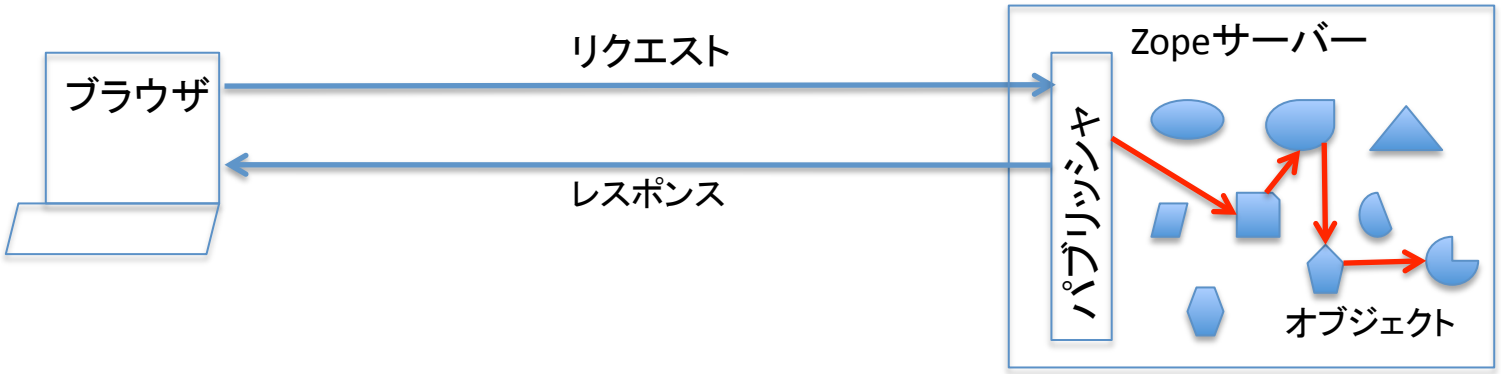
オブジェクトパブリッシング



これらのオブジェクトは、それぞれHTMLテキストや
画像データ、ブラウザで実行されるJavaScriptコード、
サーバーで実行されるPythonコードなどを含む

ブラウザからhttpリクエストが来たときに、
ファイルリソースを探してその中身を返すのではなく、
オブジェクトと呼ばれる抽象的な情報構造体を探して、
そのオブジェクトをコールして返されたものを返す。

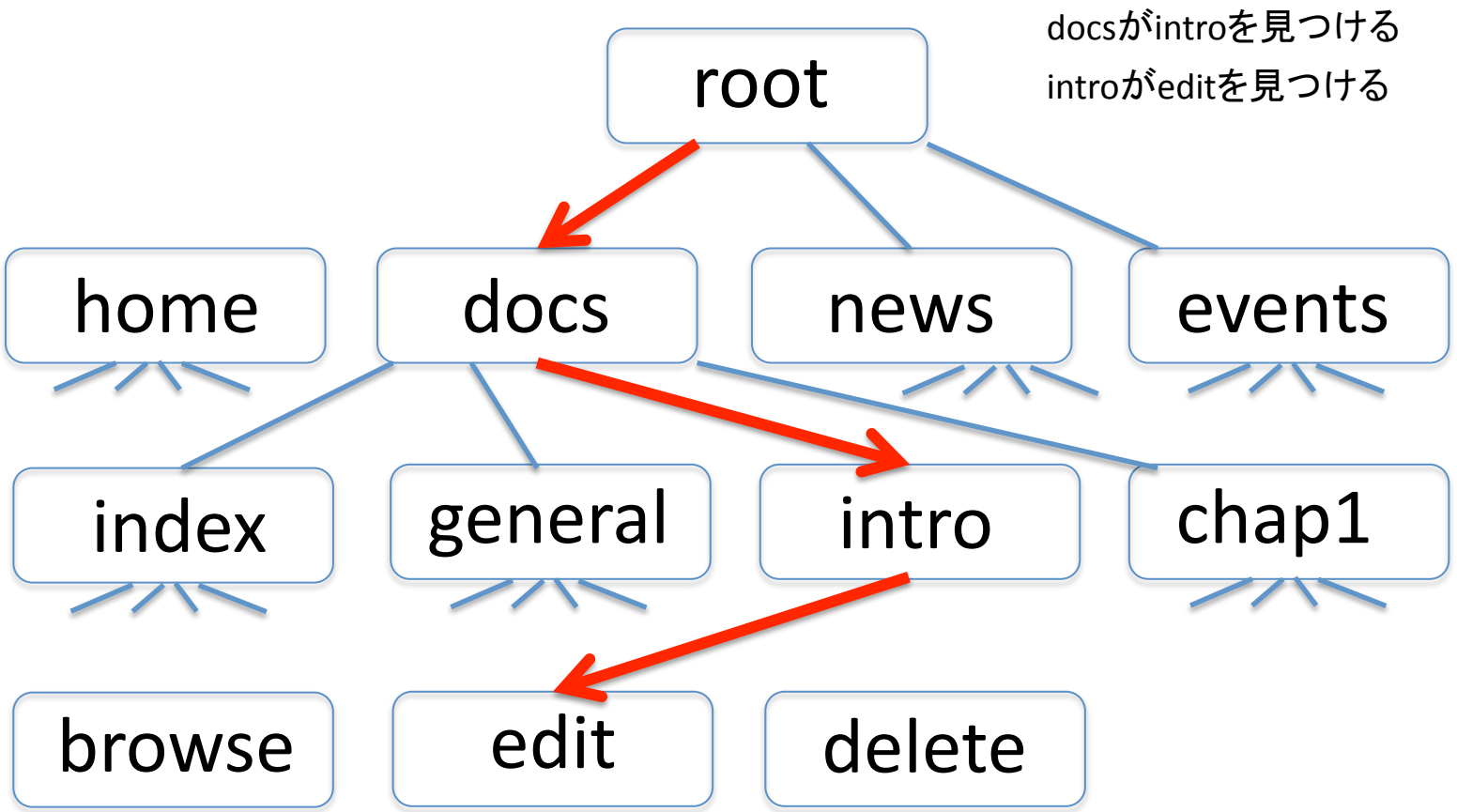
トラバーシング



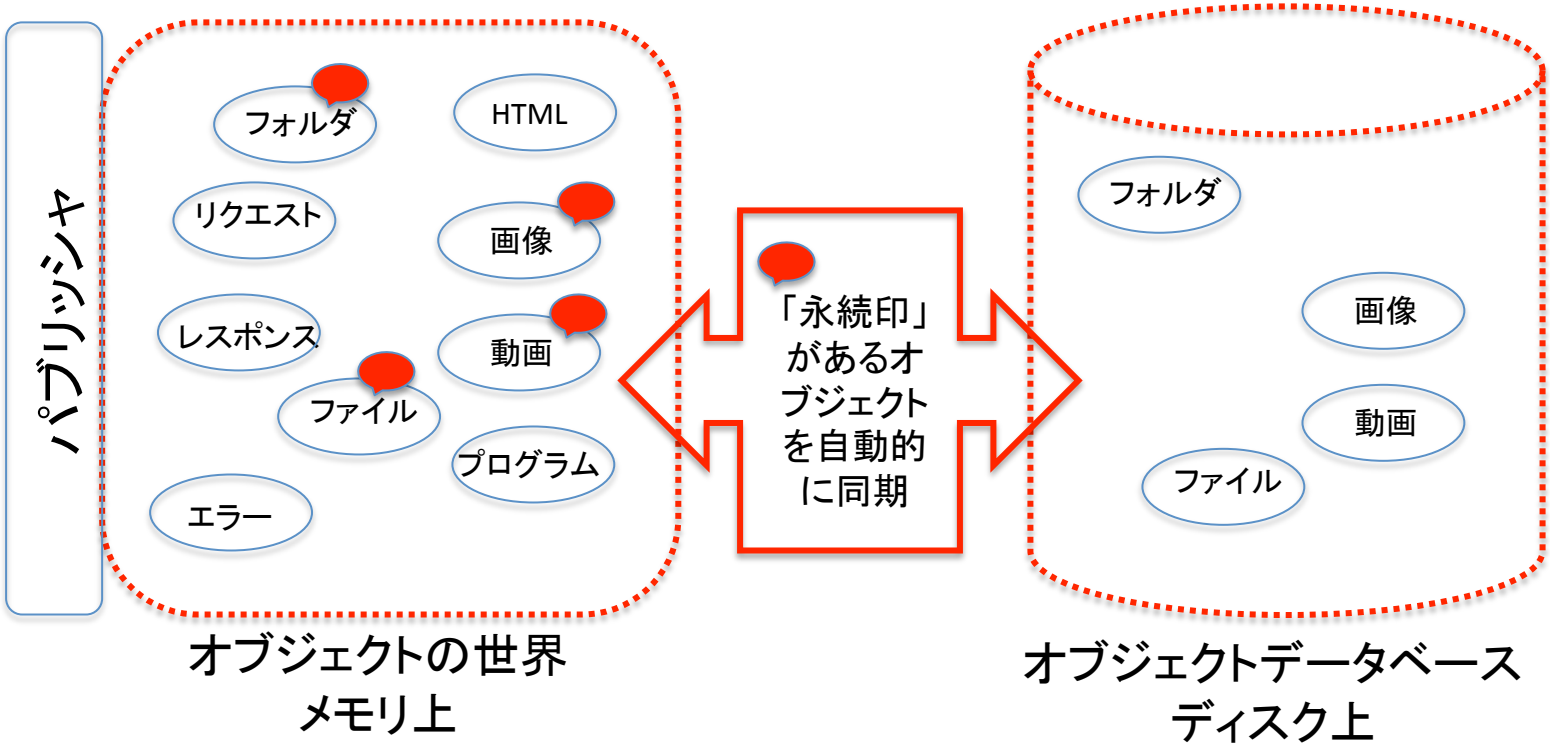
URLを受け取ると、
それを要素ごとに分解し、
それぞれの要素に対応するオブジェクトに次の要素に対応するオブジェクトを見つけてもらう。

トラバース

http://zope.jp/doc/zope/intro/edit

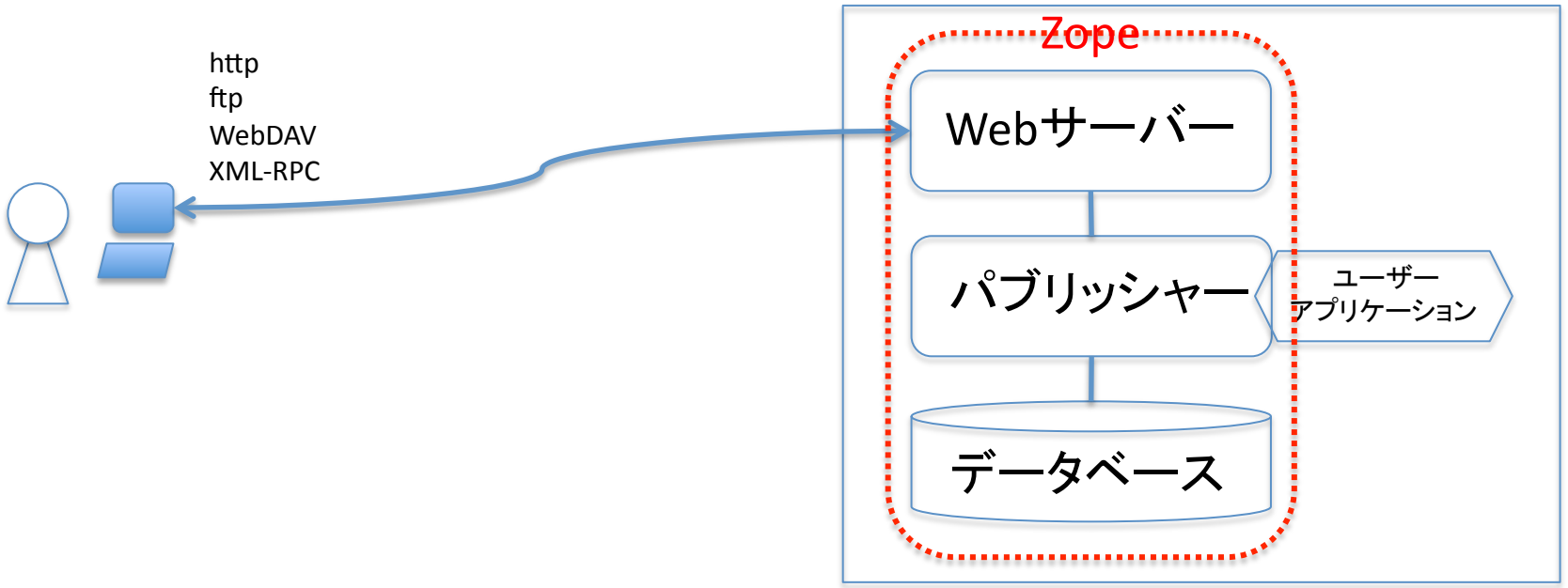


オブジェクトデータベース



永続クラスを継承することで、自動的にオブジェクトデータベース上に書き込まれるようになる。

オールインワン



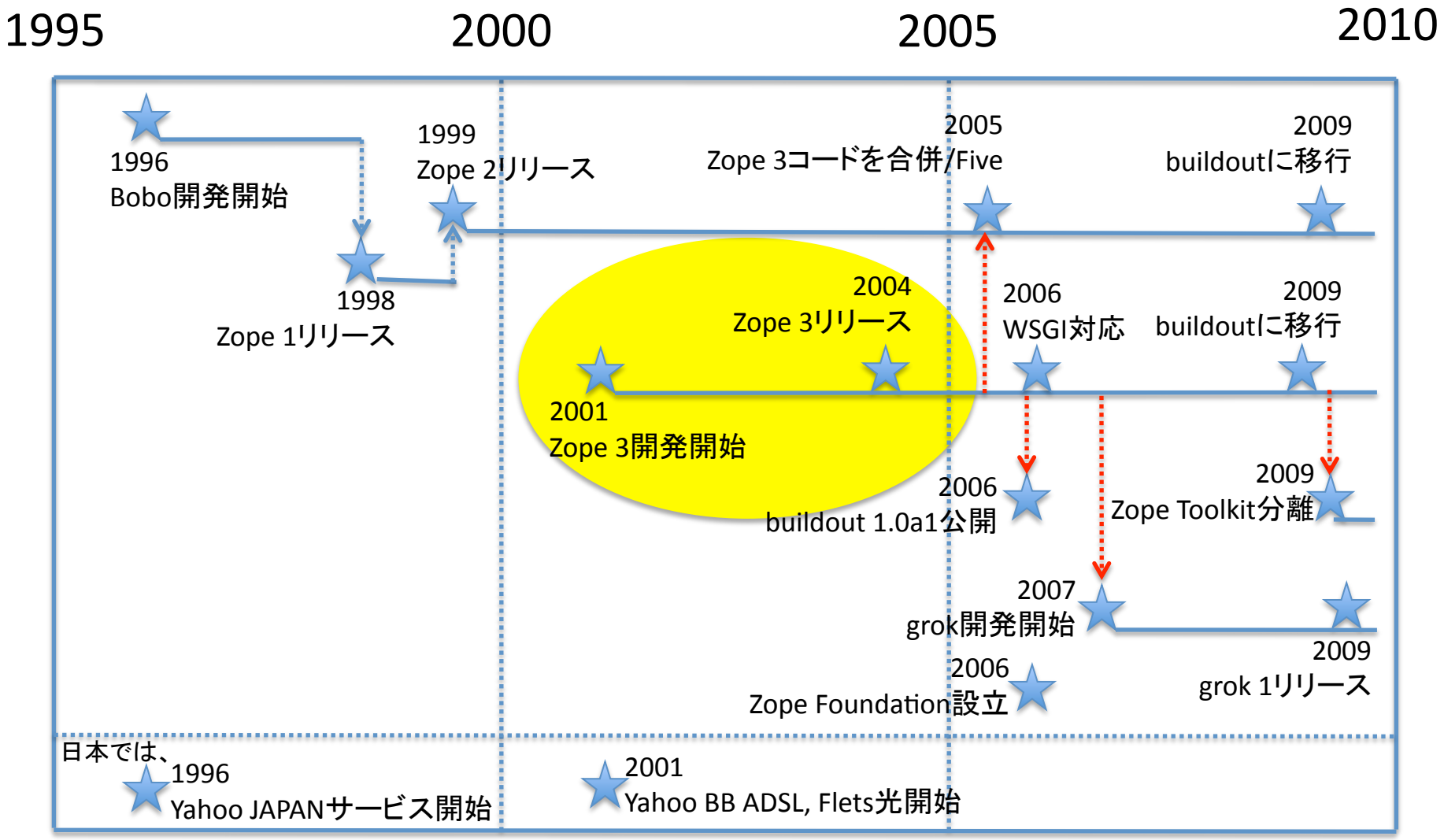
Webサーバー、パブリッシャー、データベースの
すべてが一緒に配布される。

これだけでWebアプリケーションが構築できる

オープンソース化

- 投資家の助言によってなされた
- 純粋にビジネス的な理由
  - メンテナンスの低コスト化
  - コミュニティによる継続的進化
  - ＝＞基盤部分の共同開発モデル
- この10年がこのモデルの成功を証明

2番目の波


1995
2000
2005
2010
1996
Bobo開発開始
1998
Zope 1リリース
1999
Zope 2リリース
2001
Zope 3開発開始
2004
Zope 3リリース
2005
Zope 3コードを合併/Five
2006
WSGI対応
2006
buildout 1.0a1公開
2006
Zope Foundation設立
2007
grok開発開始
2009
buildoutに移行
2009
buildoutに移行
2009
Zope Toolkit分離
2009
grok 1リリース
日本では、
1996
Yahoo JAPANサービス開始
2001
Yahoo BB ADSL, Flets光開始

Zope 3の革新

2004年 Zope 3 リリース

- コンポーネントアーキテクチャ
- テスト駆動型の開発方法

コンポーネントアーキテクチャ

既存プログラムを「部品」として再利用するための技術体系

- サブクラス化から委任へ
- インタフェース
- コンポーネントレジストリ
- ZCML
- アダプタ

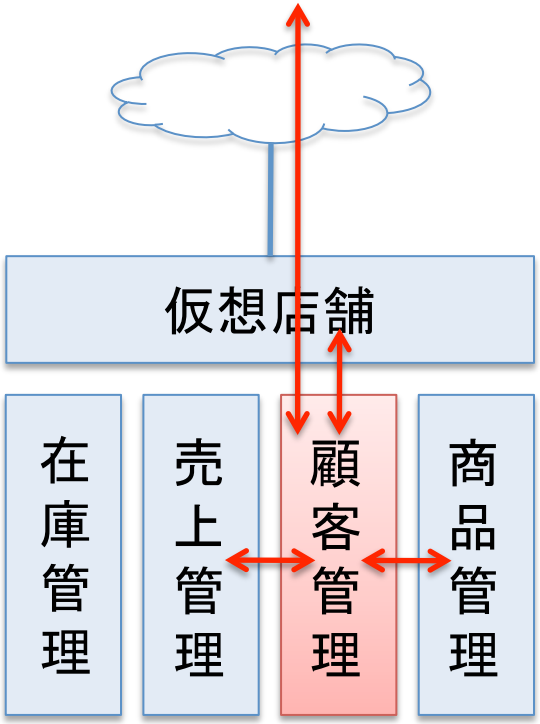
既存プログラム再利用の問題

他人が異なる目的で作った既存プログラムを利用するには、そのプログラムを改変する必要があった。

- APIの呼び方が違う
- 名前が違う
- 機能が少し違う
- セキュリティの設定が違う

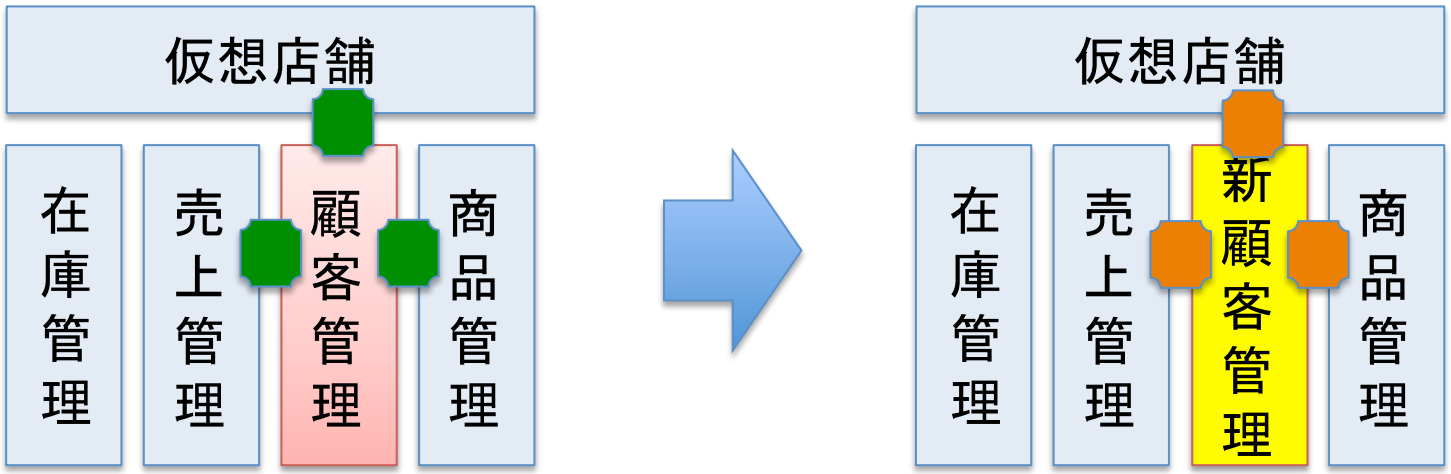
たとえば、

今ある仮想店舗の顧客管理の部分を、別の既存製品に取り換えたい場合：



- URLの要素名が違う
- APIの名前が違う
- モジュール名が違う
- 機能が少し違う
- パーミッション（セキュリティ設定）が違う

コンポーネントの交換可能性



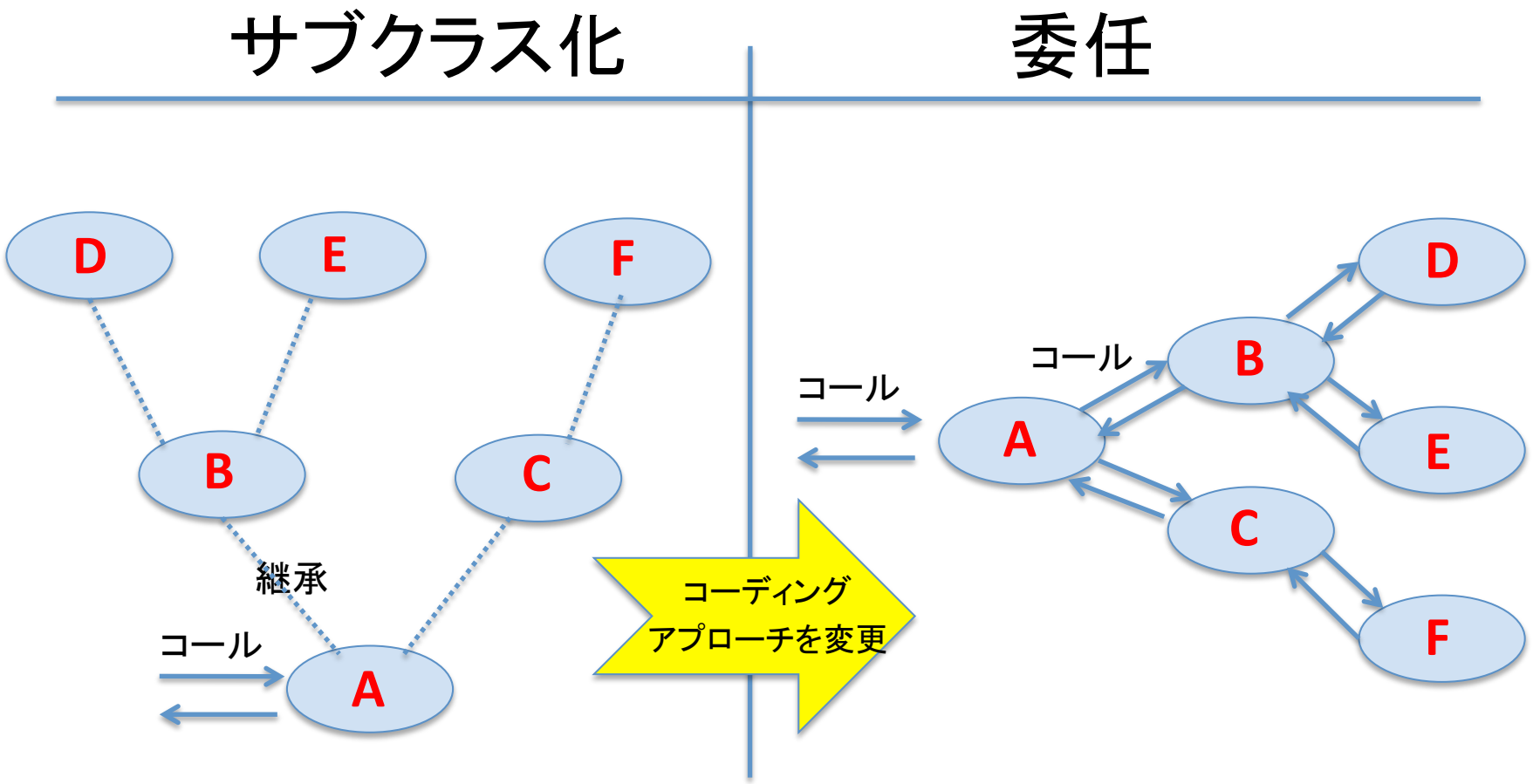
青い部分も、黄色い部分も、いっさいコードを変更しないで交換可能か？

コンポーネントアーキテクチャ

既存プログラムを「部品」として「いっさい変更せず」に「再利用」するための技術体系

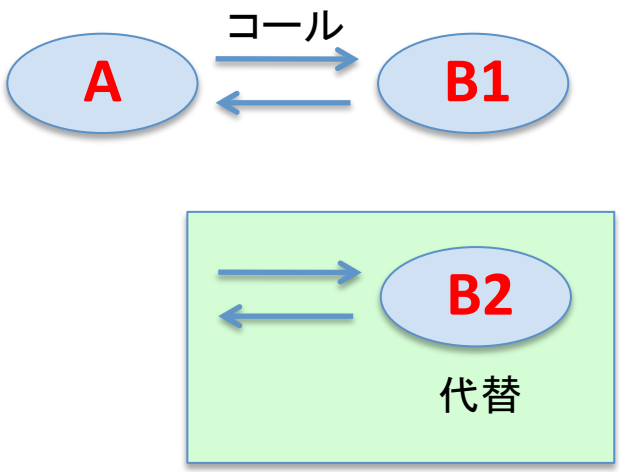
- サブクラス化から委任へ
- インタフェース
- コンポーネントレジストリ
- ZCML
- アダプタ

サブクラス化から委任へ



Aは、B, C, D, E, Fの機能を持つ

インタフェース(API定義)

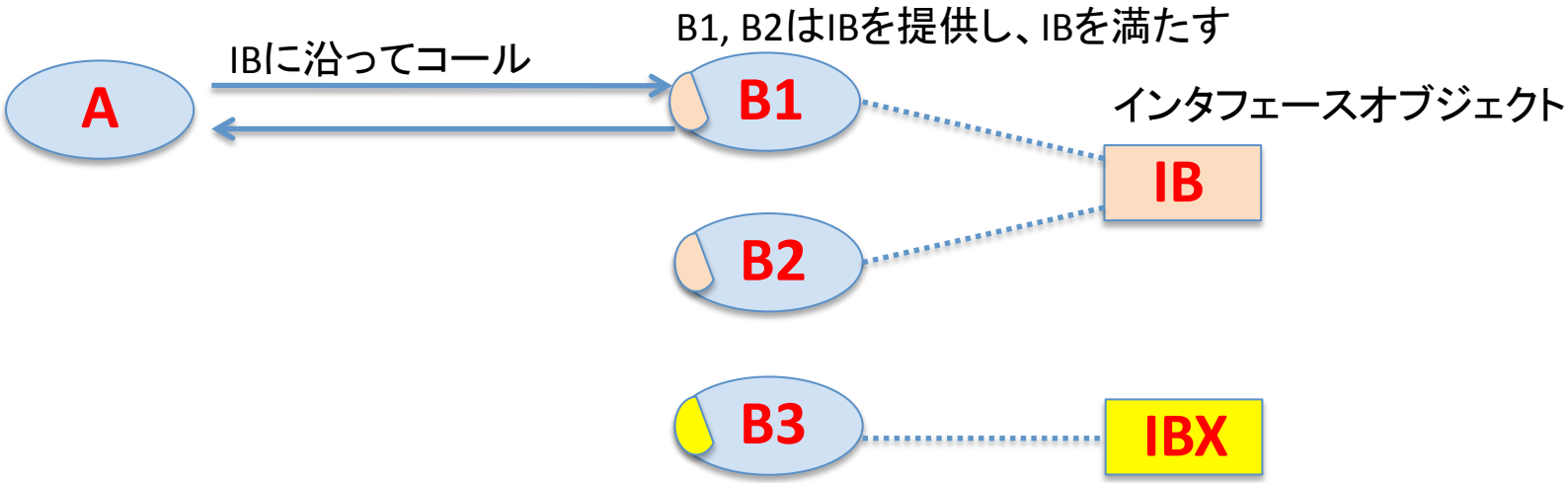


使用中のB1コンポーネントをB2コンポーネントに取り換えたい。

どうやって、B1とB1が交換可能かどうかを見分けるか？

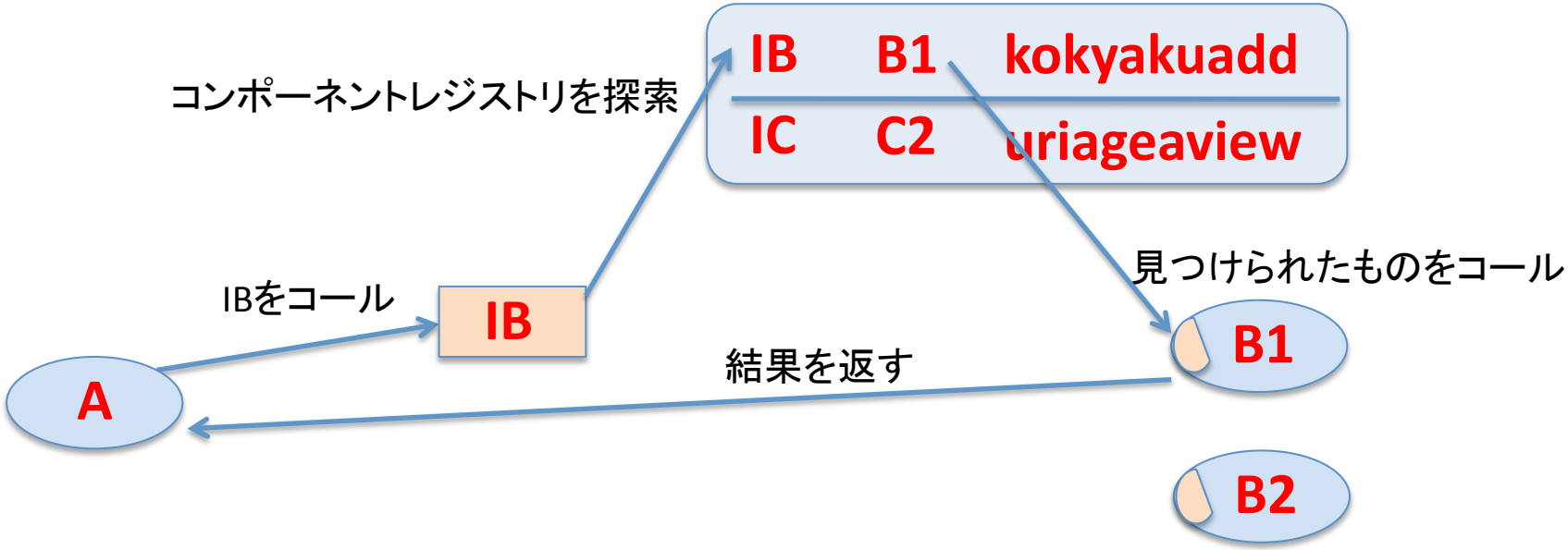
インタフェース(API定義)

コンポーネントの種類を識別するために、
インタフェースを作り、関連付ける。



呼ばれる側が同じインタフェースを提供し、それ
を満たす限り、コンポーネントとして交換可能。

コンポーネントレジストリ

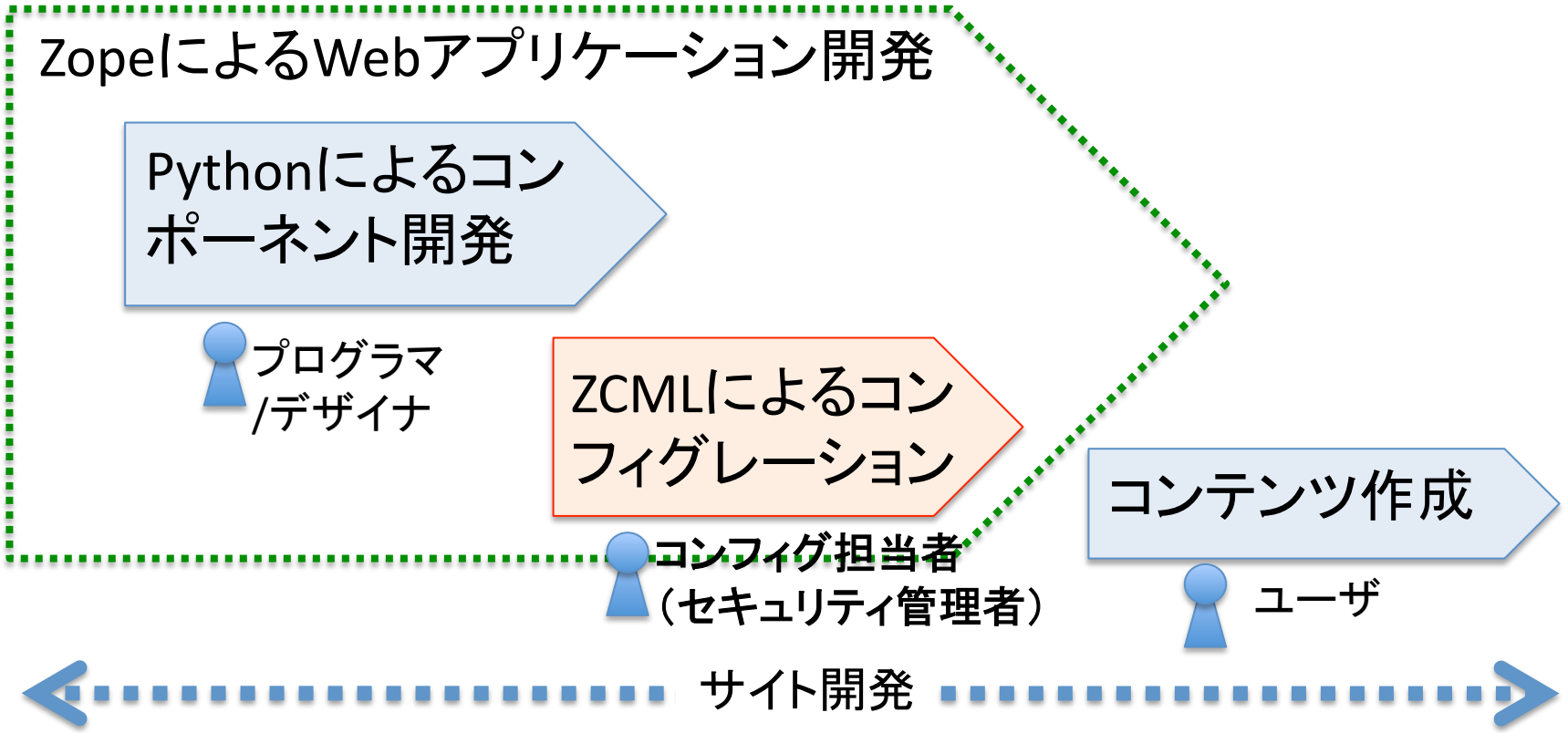


- インタフェースをキーにしてコンポーネントが登録される。
- インタフェースをコールすると、レジストリに登録されたものがコールされる。
- レジストリ登録を変えると、別のコンポーネントがコールされる。
- 登録はZCMLで行われる。
- 必要に応じてURLの要素となる名前も登録される。

ZCML

Zope Configuration Management Language

コンフィグのためのXML派生言語



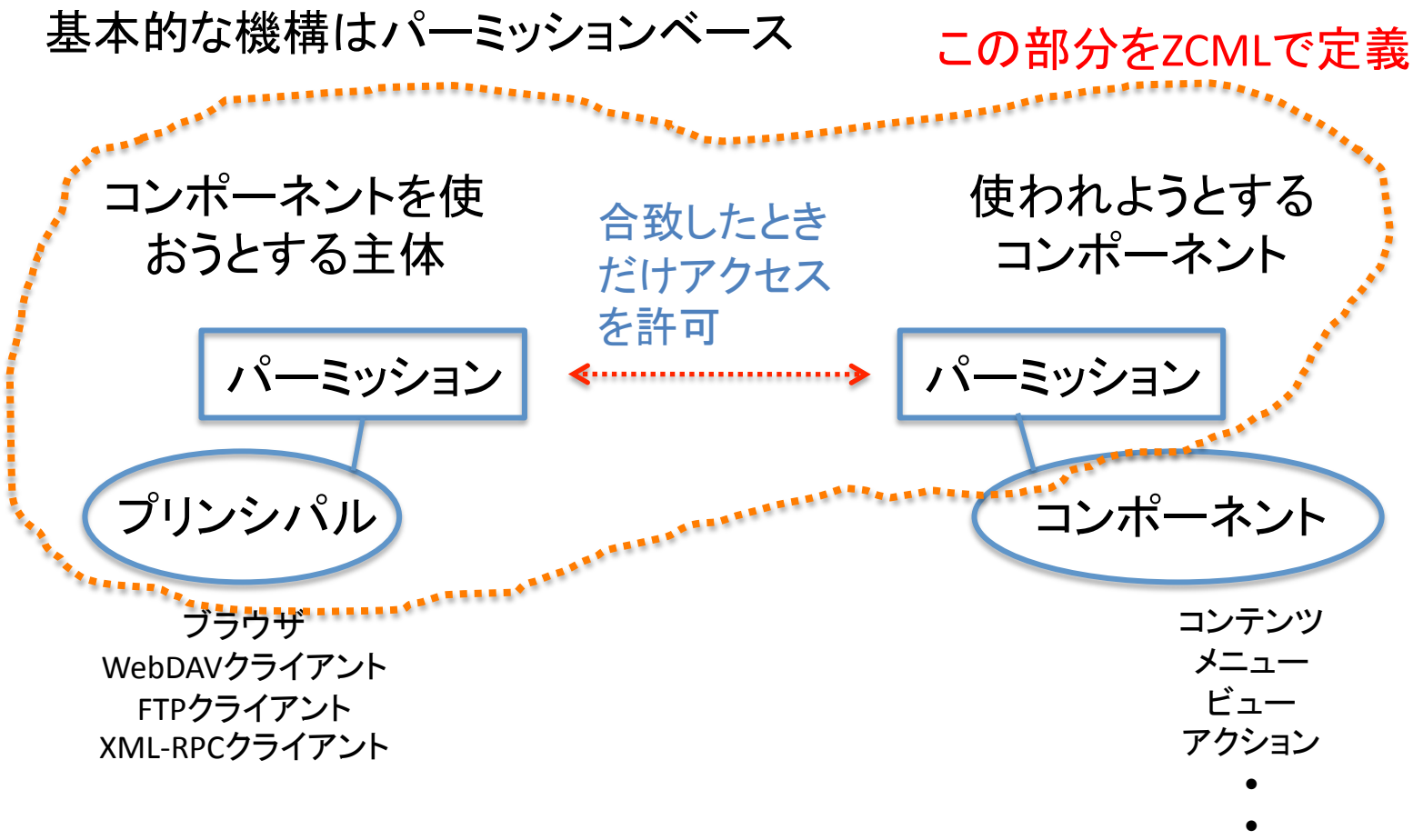
ZCMLですること

- インタフェースをキーにしてコンポーネントを登録
- URL要素に対応する名前を登録
- セキュリティ構成を定義
- メニュー構成を定義

つまり、今までPythonコードで書いていた再利用時に変更する部分を、ZCMLで記述することによって外部化した。

そして、Pythonプログラマでなくても書ける。

ZCMLによるセキュリティ定義

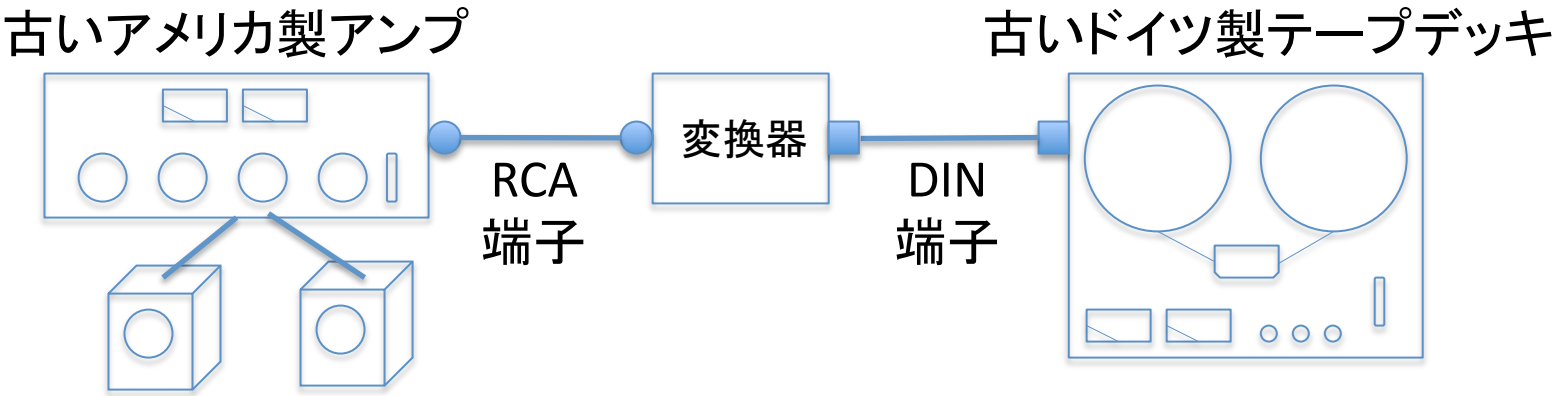


アダプタ

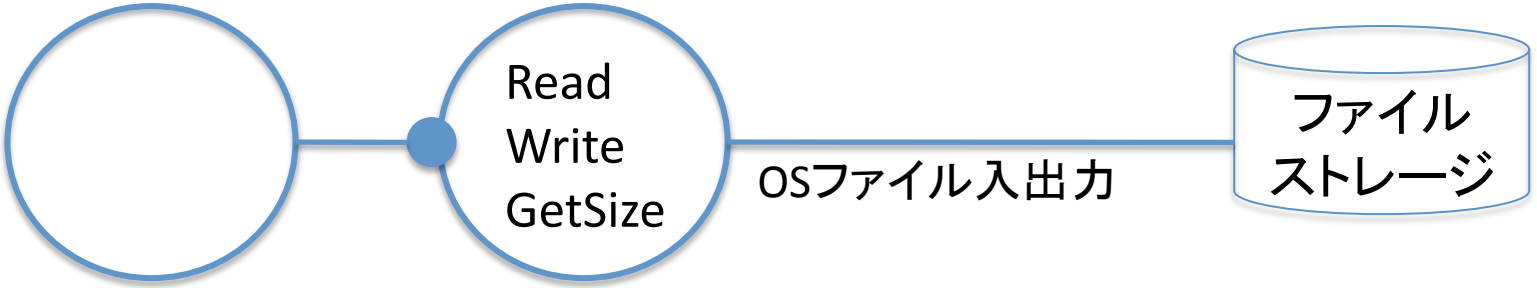
インタフェースの違いを吸収するための仲介コンポーネント

たとえるなら、

アダプタがインタフェースの違いを吸収

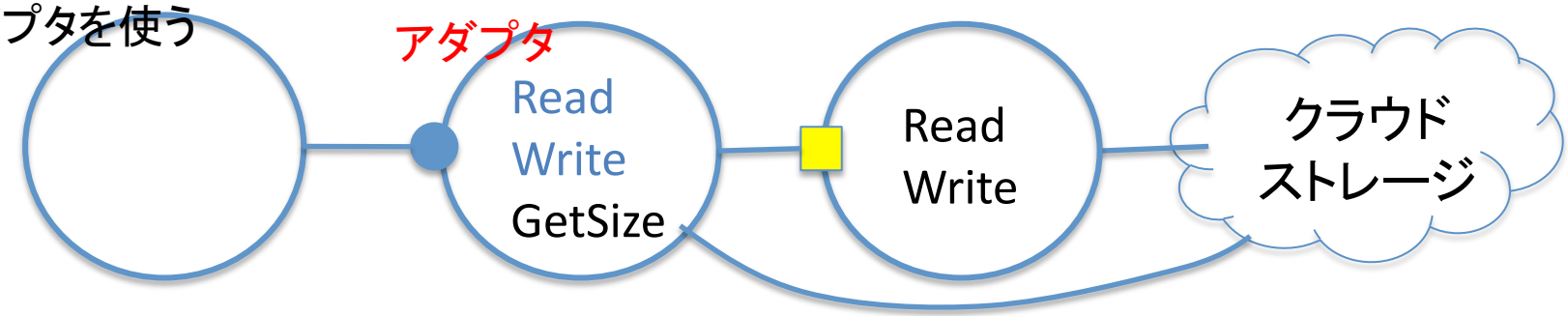


たとえば

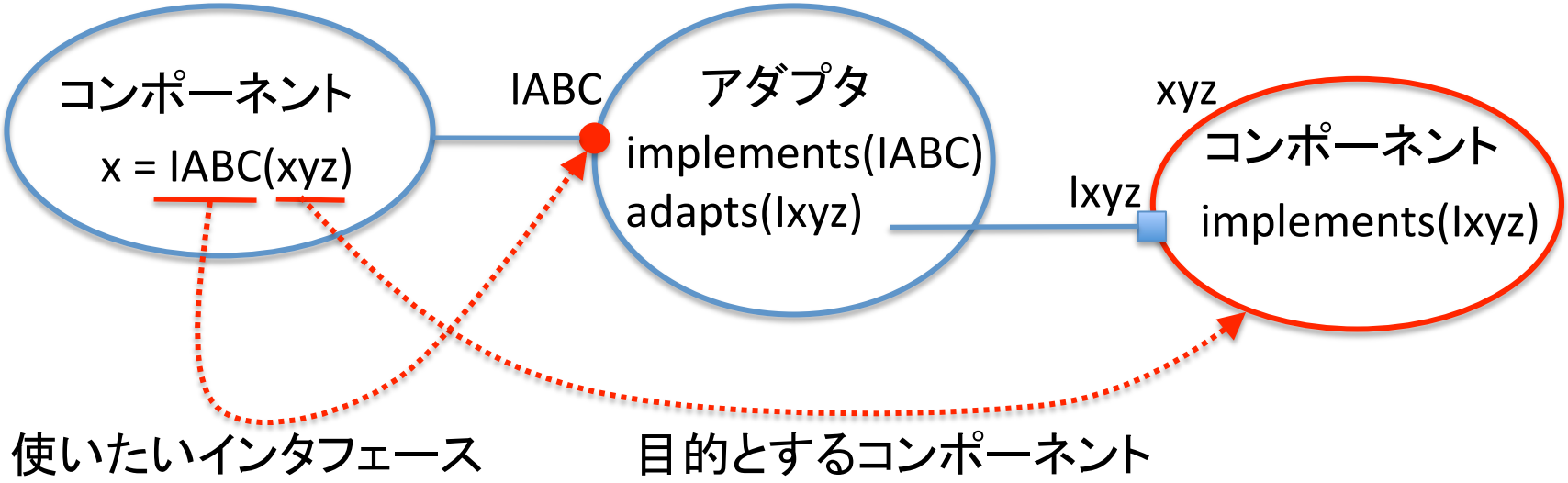


既存コードを流用してファイル入出力部分を交換したいが、インタフェースが合わない

アダプタを使う

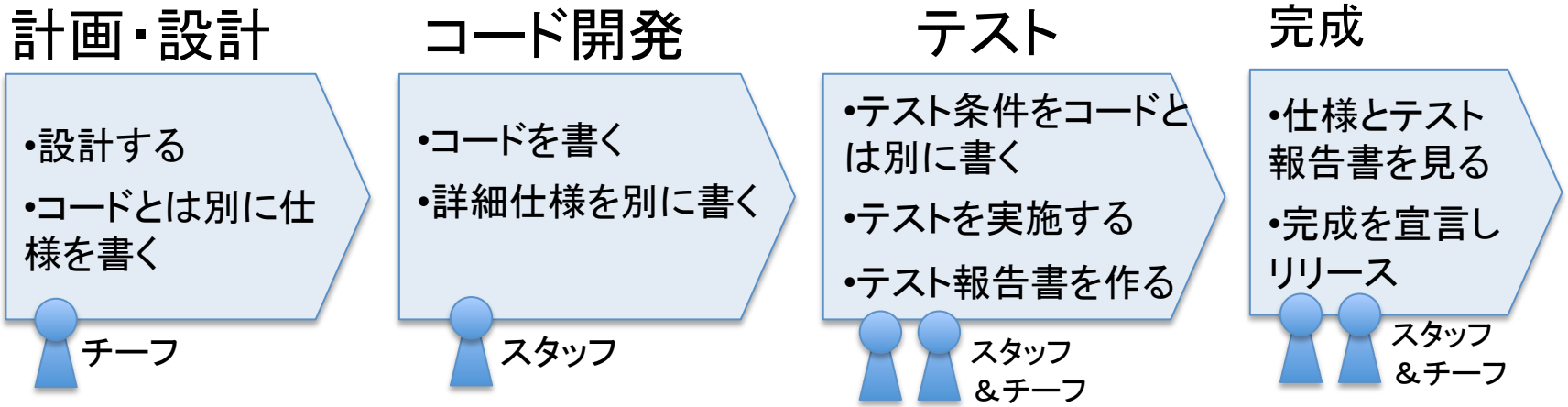


アダプタの使用

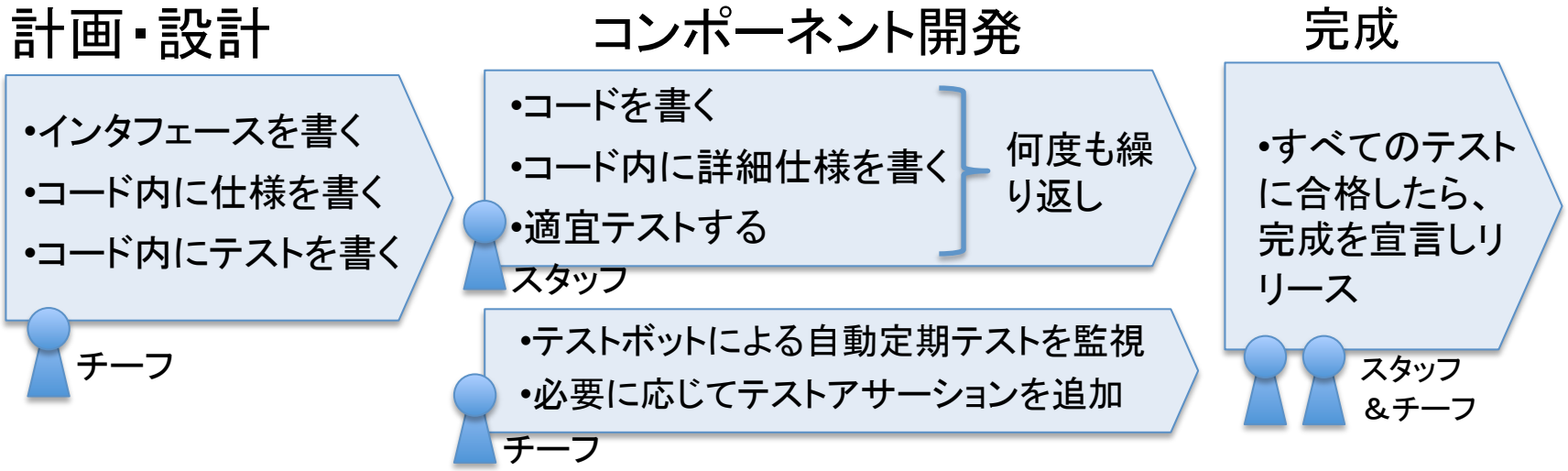

コンポーネント
x = IABC(xyz)
IABC
アダプタ
implements(IABC)
adapts(Ixyz)
xyz
Ixyz
コンポーネント
implements(Ixyz)
使いたいインタフェース
目的とするコンポーネント

テスト駆動型の開発

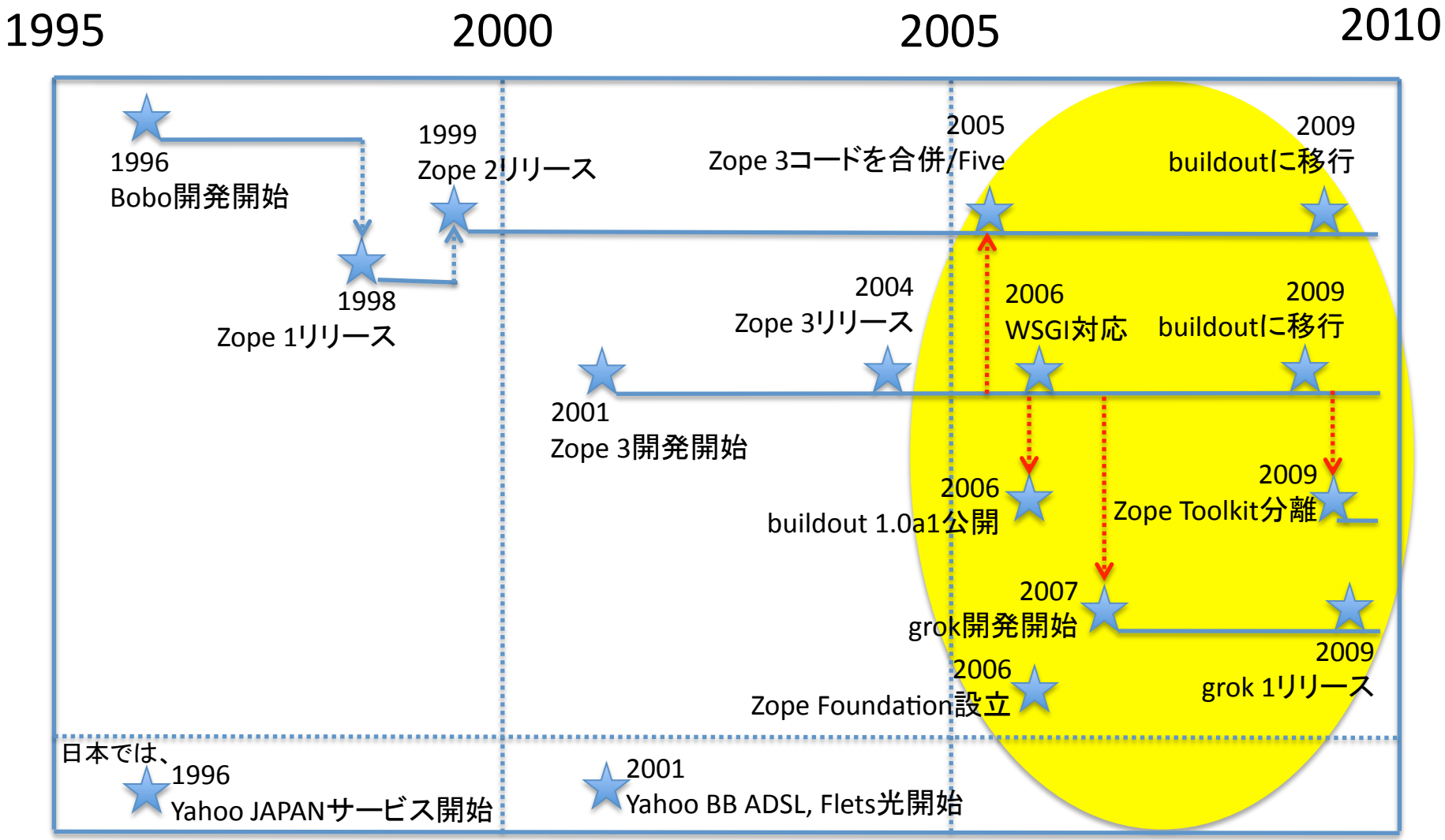
従来



テスト駆動型



現在の波

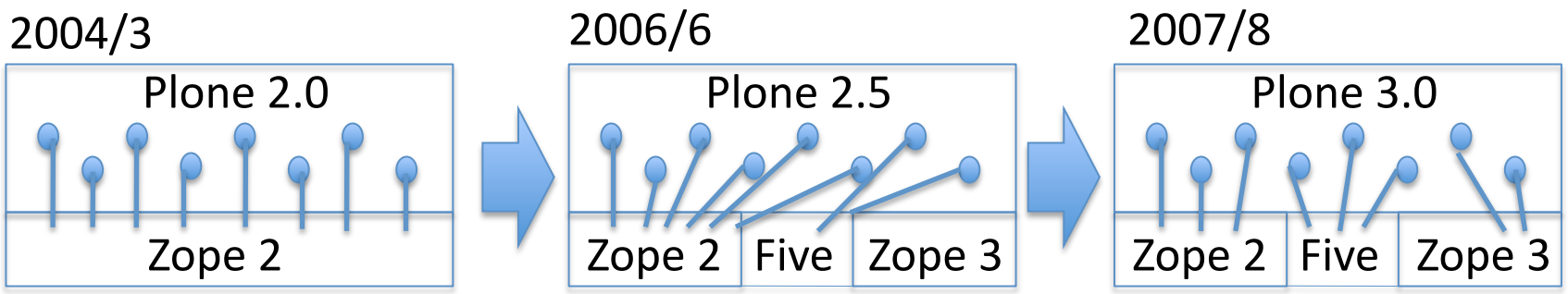

1995
2000
2005
2010
1996
Bobo開発開始
1998
Zope 1リリース
1999
Zope 2リリース
2001
Zope 3開発開始
2004
Zope 3リリース
2005
Zope 3コードを合併/Five
2006
WSGI対応
2006
buildout 1.0a1公開
2006
Zope Foundation設立
2007
grok開発開始
2009
buildoutに移行
2009
buildoutに移行
2009
Zope Toolkit分離
2009
grok 1リリース
日本では、
1996
Yahoo JAPANサービス開始
2001
Yahoo BB ADSL, Flets光開始

さらに続く革新

- Zopeコードの統合 with “Five”
- ビルドアウト
- WSGI対応

Zopeコードの統合とFive

- 2005年 Zope 2.8からZope 3とFiveを同梱
  - ZopeアプリケーションがZope 3コードを使い始める
  - 徐々に旧式なZope 2コードが廃止される

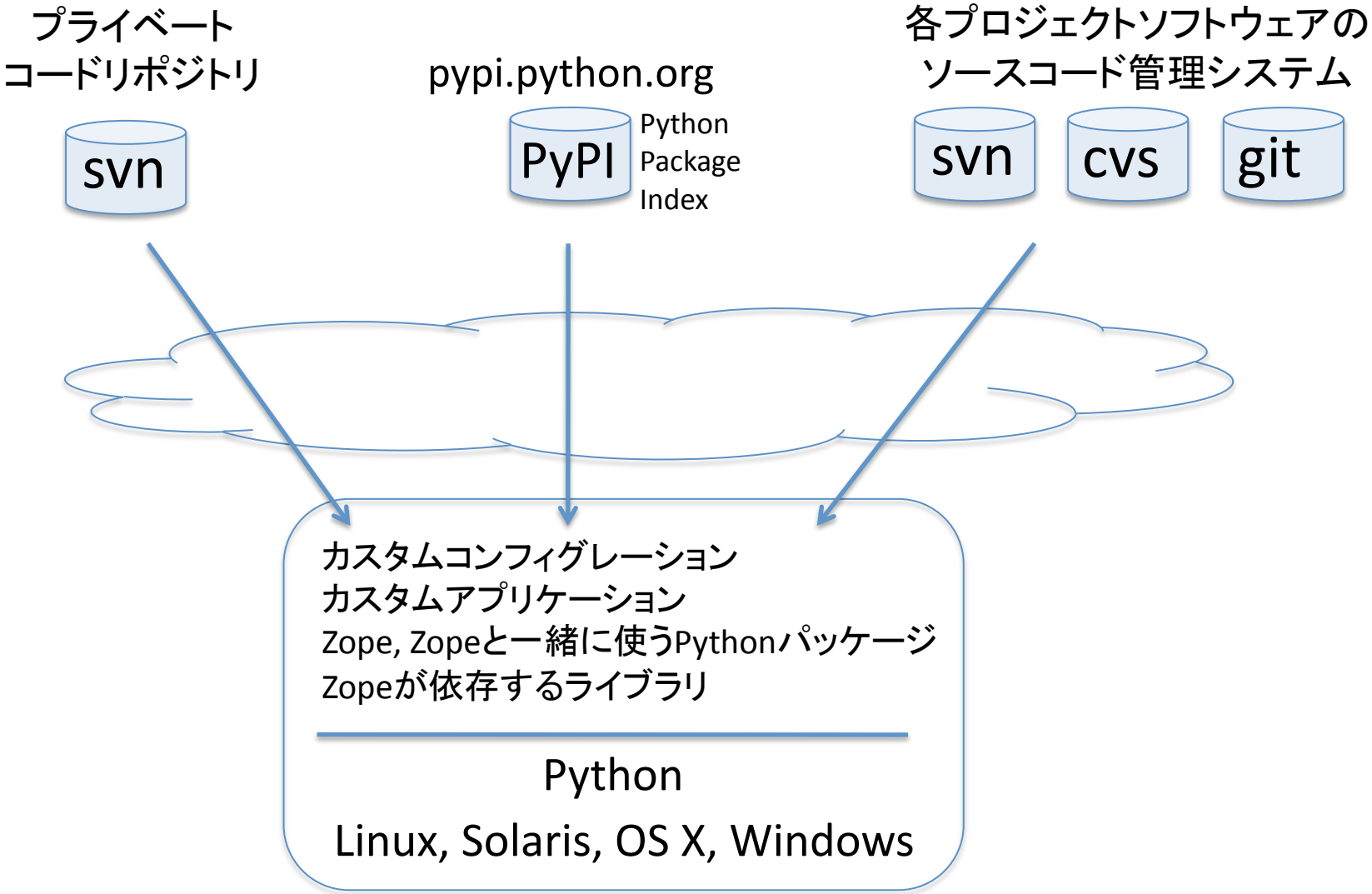


”Five”は Zope 2の中でZope 3の技術を使えるようにする技術

ビルドアウト

- デプロイメントの自動化ツール。
  - ソフトウェアのインストール
  - バージョン依存性の解決
  - 各パッケージのバージョンを手動で指定も
  - ネットワーク経由でダウンロード
  - ソースコード管理システムからも取得
  - ネットワーク設定など各種コンフィグ
  - インスタンス作成、アカウント作成
  - テスト実行

ビルドアウト


プライベート
コードリポジトリ
svn
pypi.python.org
PyPI
Python
Package
Index
各プロジェクトソフトウェアの
ソースコード管理システム
svn
cvs
git
カスタムコンフィグレーション
カスタムアプリケーション
Zope, Zopeと一緒に使うPythonパッケージ
Zopeが依存するライブラリ
Python
Linux, Solaris, OS X, Windows

なぜ？

yumやaptは、特定のLinuxでしか使えない

- Linux, Solaris, OSX, Windows,どれでも同じやり方に

makeは、コンフィグの指定単位が細かすぎる

- プログラム単位でなく、パッケージ単位で

インストールだけでなく、テスト実行やコンフィグのカスタマイズ、ユーザアカウントの作成なども自動化

デプロイメント作業

たとえば:

通常数時間

1. 必要なライブラリや関連ソフトウェアの種類とバージョンを特定し、ダウンロード、コンパイル、インストール、テスト
2. Zopeをダウンロードし、インストール、テスト
3. Zopeコンフィグレーションをカスタマイズ
4. Ploneをダウンロードし、インストール、テスト
5. Ploneコンフィグレーションをカスタマイズ
6. カスタムパッケージをインストールし、テスト
7. カスタムパッケージのコンフィグレーションをカスタマイズ
8. 初期ユーザアカウントを作成
9. 最終的なテスト

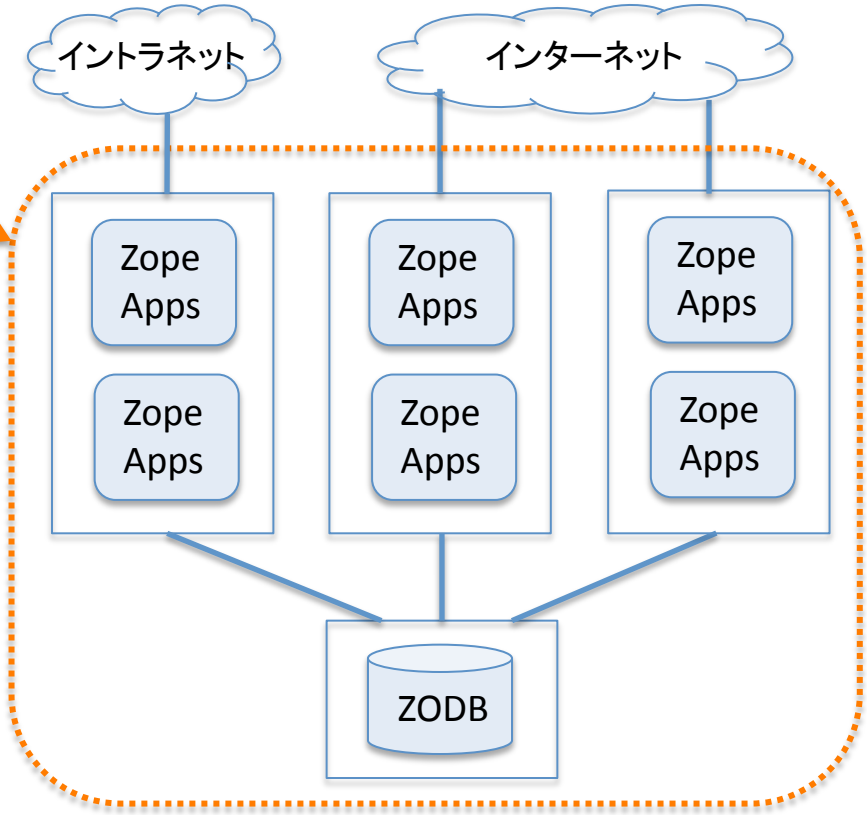
$ bin/buildout

あとは数分～数十分待つだけ

スクリプトと組み合わせて、

複数サーバの準備もコマンドひとつ

$ bin/buildout



クローンサイトの設置や、
障害復旧が迅速化

ビルドアウトはZope以外にも広がる

WSGI対応

•Python Web Server Gateway Interface

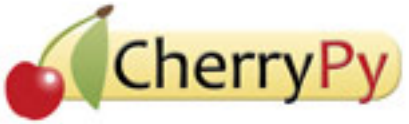
•異なるWebフレームワークが協調動作するためのAPI仕様。Java Servelet APIのPython版

•2003年 Phillip J. EbyがPEP333 として提案

•2006年 Zope 3がWSGIに対応

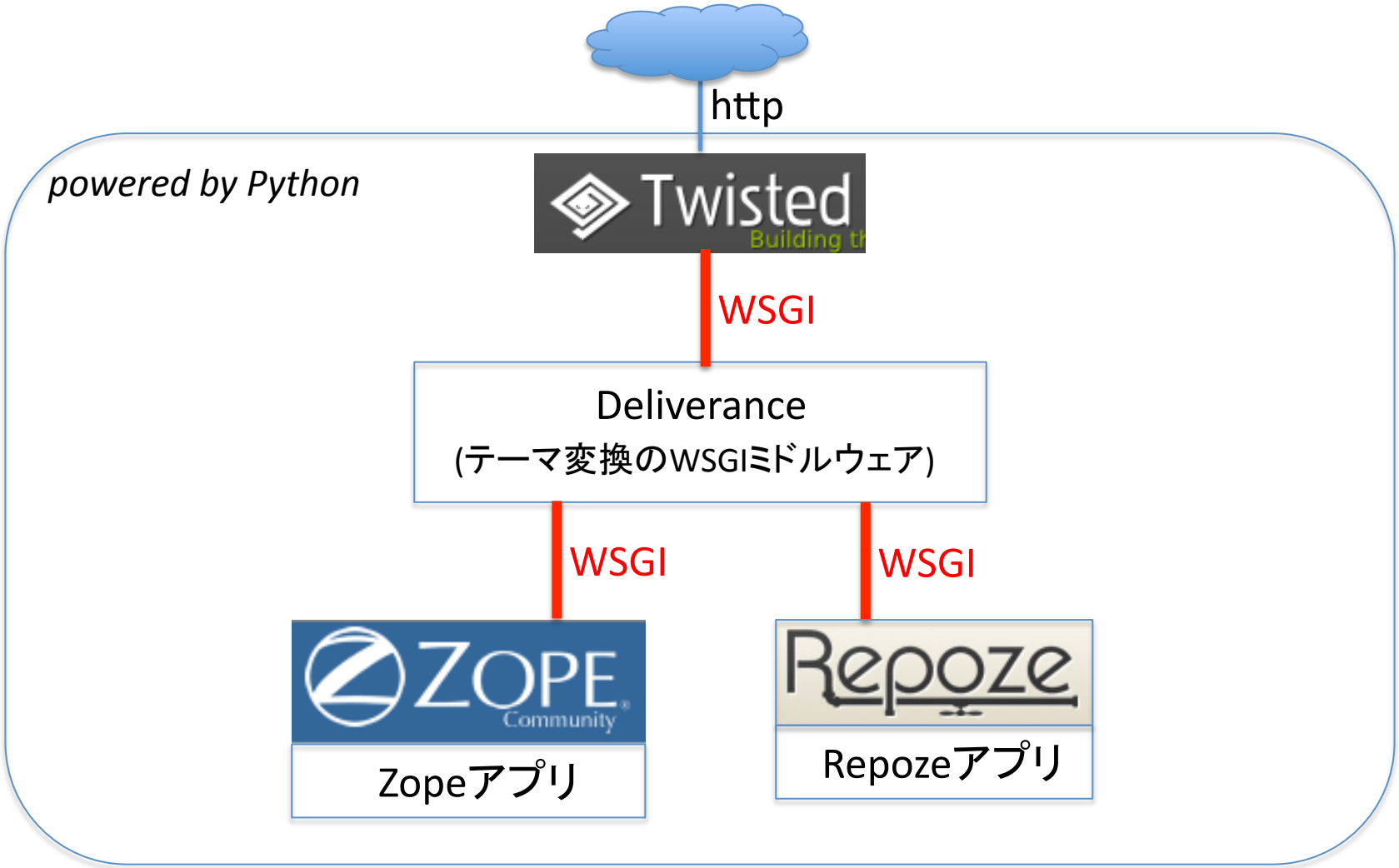
•その後、多くのフレームワークが対応

WSGIの仲間



Deliveranceなどのミドルウェアなども増加中

WSGIの使用例


http
powered by Python
Twisted
WSGI
Deliverance
(テーマ変換のWSGIミドルウェア)
WSGI
WSGI
ZOPE
Community
Zopeアプリ
Repoze
Repozeアプリ

Zopeパッケージ形態の多様化

Zopeユーザ（Webアプリケーションの
開発者）の使いやすさを追求すると、

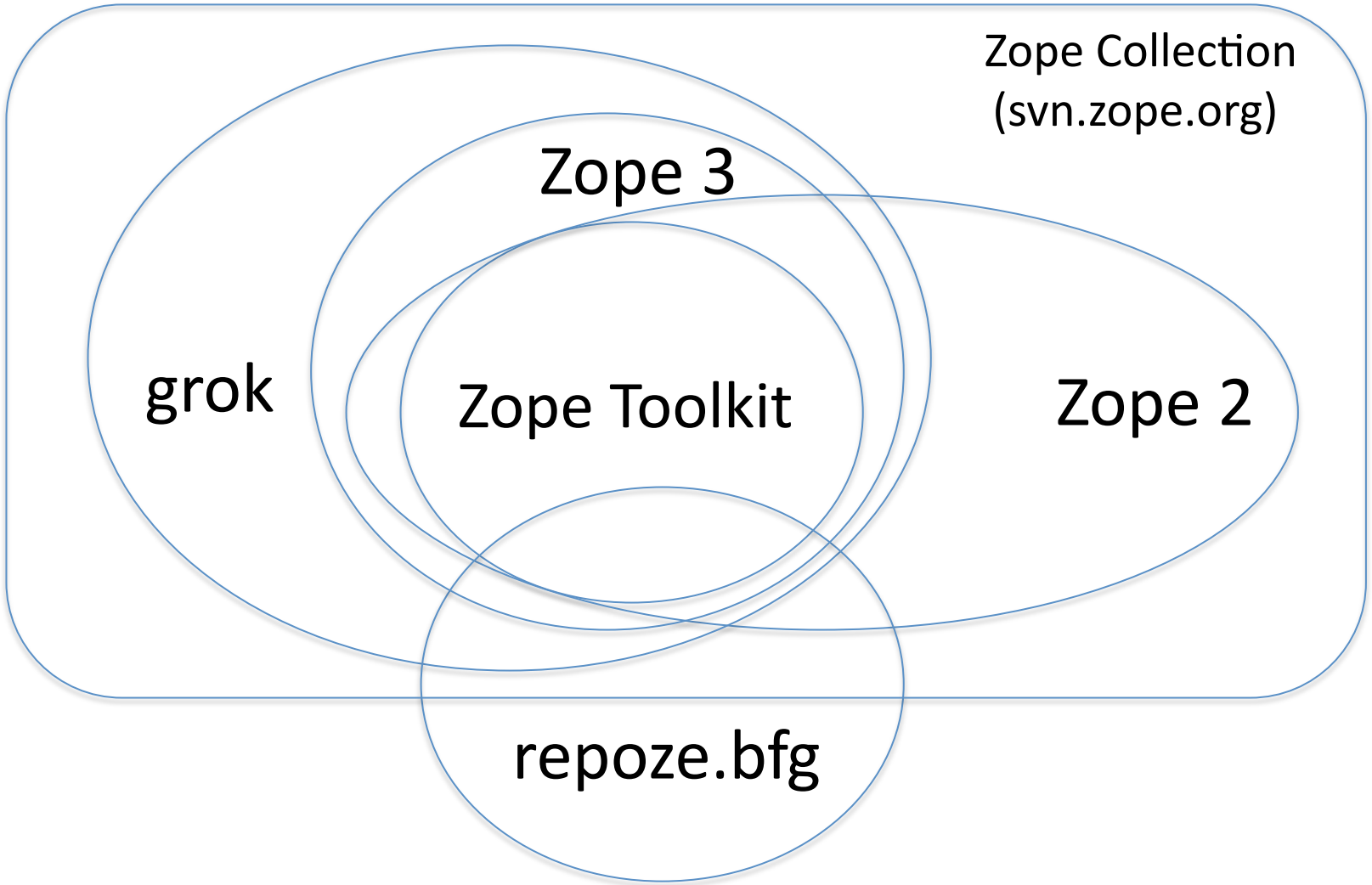
プラットフォーム（大きな単体基盤）

フレームワーク（全体でも部分でも使える基盤）

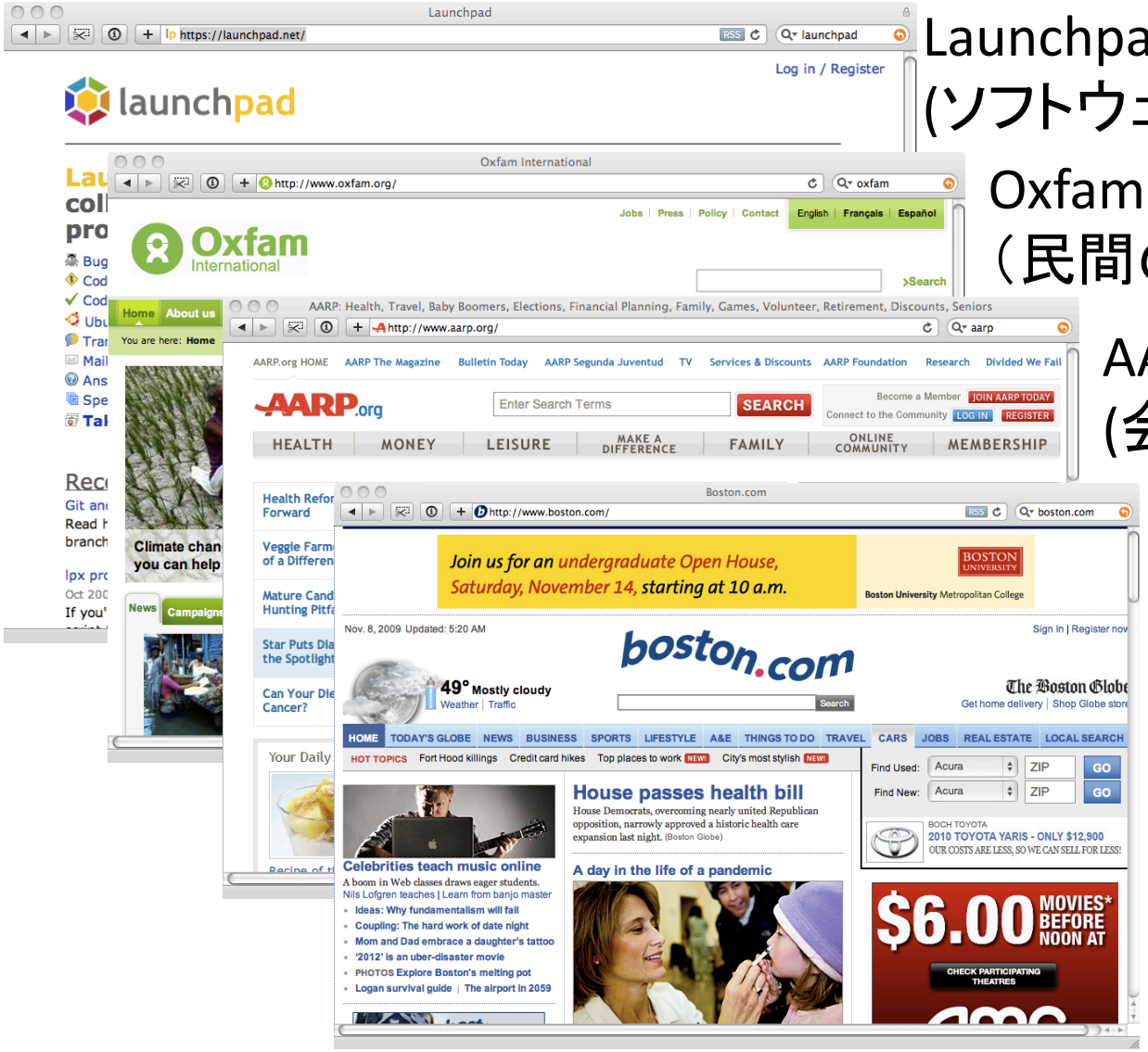
ライブラリ（選択可能な部品の集合）

KGS(Know Good Set)を集めて大きなフレームワークが作れる。

Zopeコード利用の拡大



Zopeを使ったサイト



Launchpad
(ソフトウェア協働基盤)

Oxfam (Int'l, America, UK)
（民間の国際協力団体）

AARP
(会員3000万人のNPO)

boston.com
（新聞社）

Zopeを使った.govサイト

NOAA (米国海洋大気庁)

NASA (米国航空宇宙局)

CIA (米国中央情報局)

FBI (米国連邦捜査局)

Zopeを使った公共系サイト

Spring-8

大阪大学

ジェトロ

山形県

まとめ

- 1998年 **オブジェクト指向技術**を使い、複雑なWebアプリケーションの開発を飛躍的に効率化。
- 2004年 **コンポーネントアーキテクチャ**を使い、既存プログラムを再利用できるようにした。
- 2006年 **ビルドアウト**を使って、デプロイメントを飛躍的に効率化。
- Zope技術は、Zope以外のソフトウェアにも広まりつつある。

Zopeをもっとよく知る

日本Zopeユーザー会：http://www.zope.jp
同メーリングリスト　　：http://ml.zope.jp
グローバルコミュニティ：http://www.zope.org

Zope 3 入門篇

発展篇も近日発売

これが原著

トレーニングとコンサルティング
info@iccm.jp