

アプリ開発を楽にする Ruby PaaS 「MOGOK」のご紹介

株式会社インターネットイニシアティブ
プラットフォームサービス部プラットフォーム開発課

阿部 博 <abe@iij.ad.jp>

2013/11/19

本セッションの概略

- IIJが提供するPaaSであるMOGOKを紹介します。MOGOK/PaaSを利用することにより何がどのように楽に/便利になるのかを技術的側面を含め解説します。

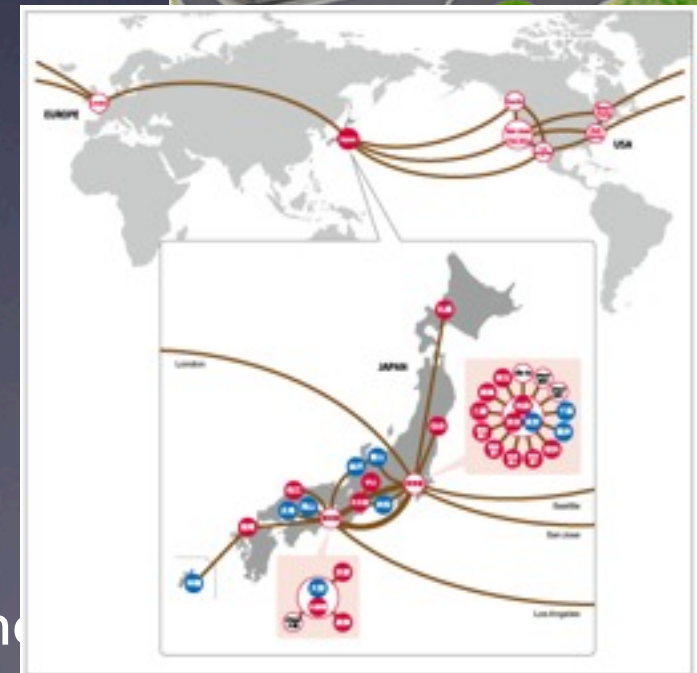
発表内容

- IJとPaaS
- MOGOKについて
- PaaSの本質
- MOGOKのシステム
- MOGOKの今後

IIJ と PaaS

IIJの強み

- 自社回線 (Internet, Backbone, Mobile)
- 自社データセンター
- 自社クラウド (IIJ GIO)
- セキュリティ



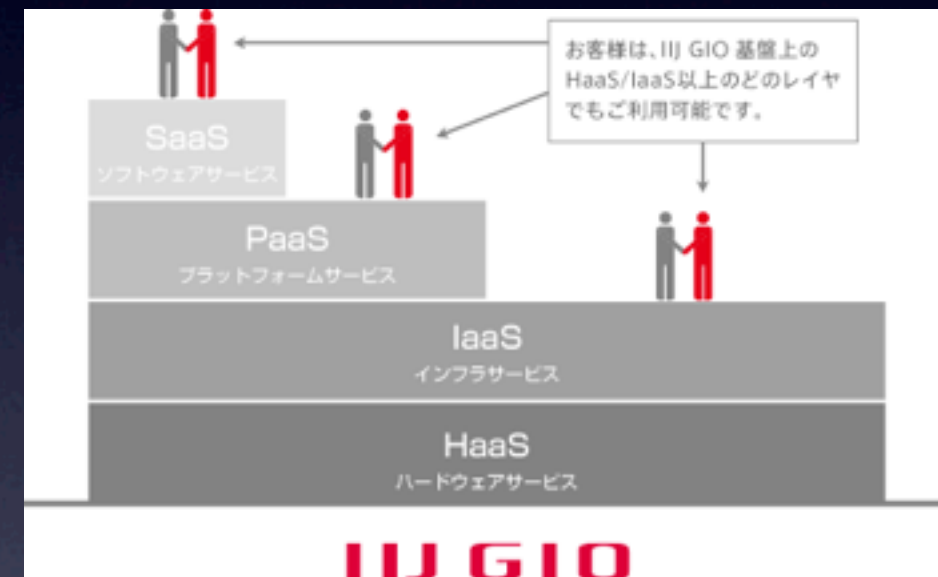
IJとクラウド

- IaaSは既にできている（それなりに売れている）

- IJ GIO

- SaaSもそれなりに売れている

- 間を取ってPaaSは？



- そもそも日本に市場が余り存在しない
 - トライする価値あり（市場を作る意味でも）

そもそもそもそもPaaSとは？

- Platform as a Service
 - プラットホームを提供するサービス
- IaaSよりも運用管理が楽
- 開発に注力ができる
- お手軽なデプロイ方法
- スケールアップ／スケールアウトの容易性

PaaSの市場規模

- IDCのPaaS市場規模予想
 - 国内PaaS市場(2012年)
 - 前年比63.4%増の263億円と予想
 - 2016年の市場規模で1,070億円
 - ちなみにIaaS市場は...(2012年)
 - 前年比49.6%増の316億円と予想
 - 2016年の市場規模で1,035億円

PaaSの市場規模

- IDCのPaaS市場規模予想
 - 国内PaaS市場(2012年)
 - 前年比63.4%増の263億円と予想
 - 2016年の市場規模で1,070億円
 - ちなみにIaaS市場は...(2012年)
 - 前年比49.6%増の316億円と予想
 - 2016年の市場規模で1,035億円

まだ伸びしろ
がある！！

MOGOKについて

MOGOKについて

- <http://mogok.jp/>



お時間がある方は
ここをクリック:-)

名前の由来



MOGOKについて

- サービス内容



- Ruby on Railsのアプリを動作させる
WebインスタンスとDB、その他の
機能／環境を提供（PaaS）
- 環境占有型ではなく共用型

ポータル

MOGOK

IIJ
Internet Initiative Japan

ようこそ、abeさん 一般

ログアウト

ポータルトップ

システムメニュー

アプリケーション

ダウンロード

アカウント

アカウント管理

退会申請

サポート

ドキュメント

FAQ

掲示板

問い合わせ

アプリケーション一覧

アプリケーション

検索

登録

アプリケーション名	URL	操作	作成日時	
hirolovesbeer	http://hirolovesbeer.ruby.iijgio.com/	  	2012/10/29 13:36:54	 
hirolovesbeer3	http://hirolovesbeer3.ruby.iijgio.com/	  	2013/07/08 14:59:31	 

©2011-2013 Internet Initiative Japan Inc.

© 2013 Internet Initiative Japan Inc.

mogokコマンド

- 20個ほどのコマンドオプションがある
- 主に以下の3種類
 - アプリケーションを管理するコマンド
 - サーバの設定を行うコマンド
 - 開発・保守・調査のためのコマンド

アプリの起動まで

- コードを書いて必要なコマンドを打つと...

```
$ cd hogegege
```

```
$ git init
```

```
$ git add .
```

```
$ git commit -m 'first commit'
```

```
$ mogok create fugafuga
```

```
$ git push mogok master
```

```
$ mogok build
```

```
$ mogok start
```

アプリの起動まで

- アプリが起動する！！

The screenshot shows the Redmine interface. At the top, there are navigation links: ホーム, マイページ, プロジェクト, 管理, ヘルプ. On the right, it says ログイン中: admin, 個人設定, ログアウト. Below the navigation is a search bar with the text 検索: and a search input field. The main content area is titled プラグイン. It displays a table with one entry:

プラグイン	開発者	バージョン
Redmine Wiki Extensions plugin This is a Wiki Extensions plugin for Redmine http://www.r-labs.org/projects/r-labs/wiki/Wiki_Extensions_en	Haruyuki Iida	0.6.1

On the right side of the page, there is a sidebar menu titled 管理. The menu items are: プロジェクト, ユーザー, グループ, ロールと権限, トラッカー, チケットのステータス, ワークフロー, カスタムフィールド, 列挙項目, 設定, LDAP認証, プラグイン (highlighted), and 情報.

ソースコードの保存

- MOGOKのgitリポジトリに保存

```
$ cd hogehoge
```

```
$ git init
```

```
$ git add .
```

```
$ git commit -m 'first commit'
```

```
$ mogok create fugafuga
```

```
$ git push mogok master
```

```
$ mogok build
```

```
$ mogok start
```

MOGOKについて

- 簡単に登録できる！！
- 簡単に環境セットアップができる！！
- 簡単にアプリを動かせる！！

PaaSの本質

なぜPaaSか？

- アプリ開発を楽に！！
 - 開発者が開発に専念
 - 運用からの解放

なぜPaaSか？

- 開発者が開発に専念
 - コード記述に集中できる
 - お手軽なデプロイツール／デプロイ方法を利用できる
- 運用からの解放
 - 環境（VM、ハードウェア）の調達／運用
 - ソフトウェアの管理（Webサーバ、DBなど）
 - セキュリティ対策

PaaSの本質

- ユーザ建前
 - 「開発に専念させて下さい」

PaaSの本質

- ユーザ本音
 - 「開発に専念させて下さい」
 - 「運用を肩代わりして下さい」

PaaSの本質

- 運用をPaaS事業者へとアウトソース
- 運用コストを払って開発に専念

アプリ開発を楽に！！

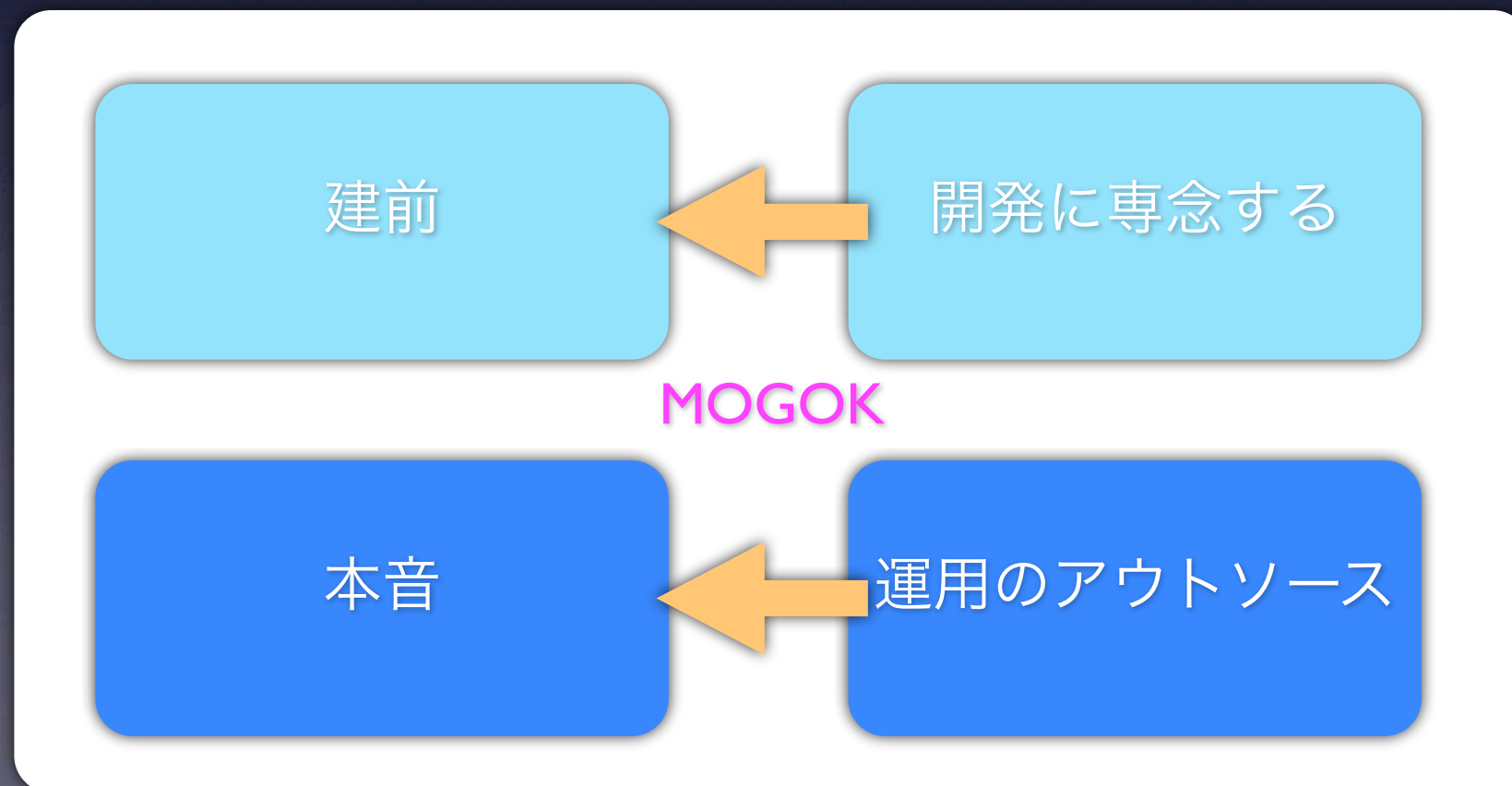
- ソースコードはgitサーバで管理
- buildの自動化
- Webサーバも提供される
- DBサーバも提供される
- 処理が重くなったらスケールアップ/スケールアウトできる

アプリ開発を楽に！！

- 運用はIJにお任せ
 - 24時間/365日の運用をアウトソース
- サーバ/アプリケーションのセキュリティ対策もIJにお任せ
 - 年に数回から数十回の対応

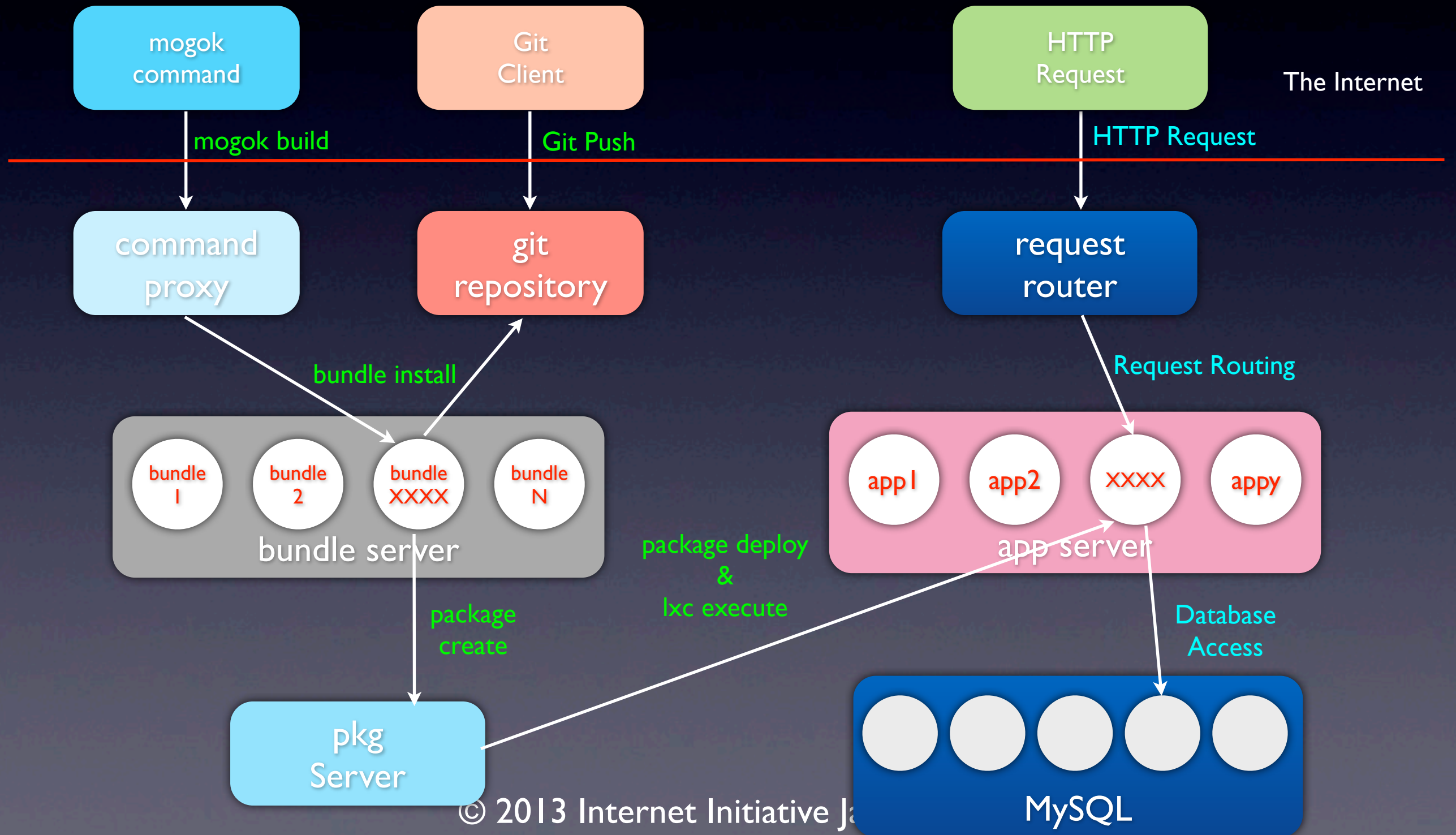
本音と建前の実現

- MOGOK = PaaSの本質を実現する環境
- PaaSの本質 = 本音と建前の両方の実現



MOGOKのシステム

システムイメージ

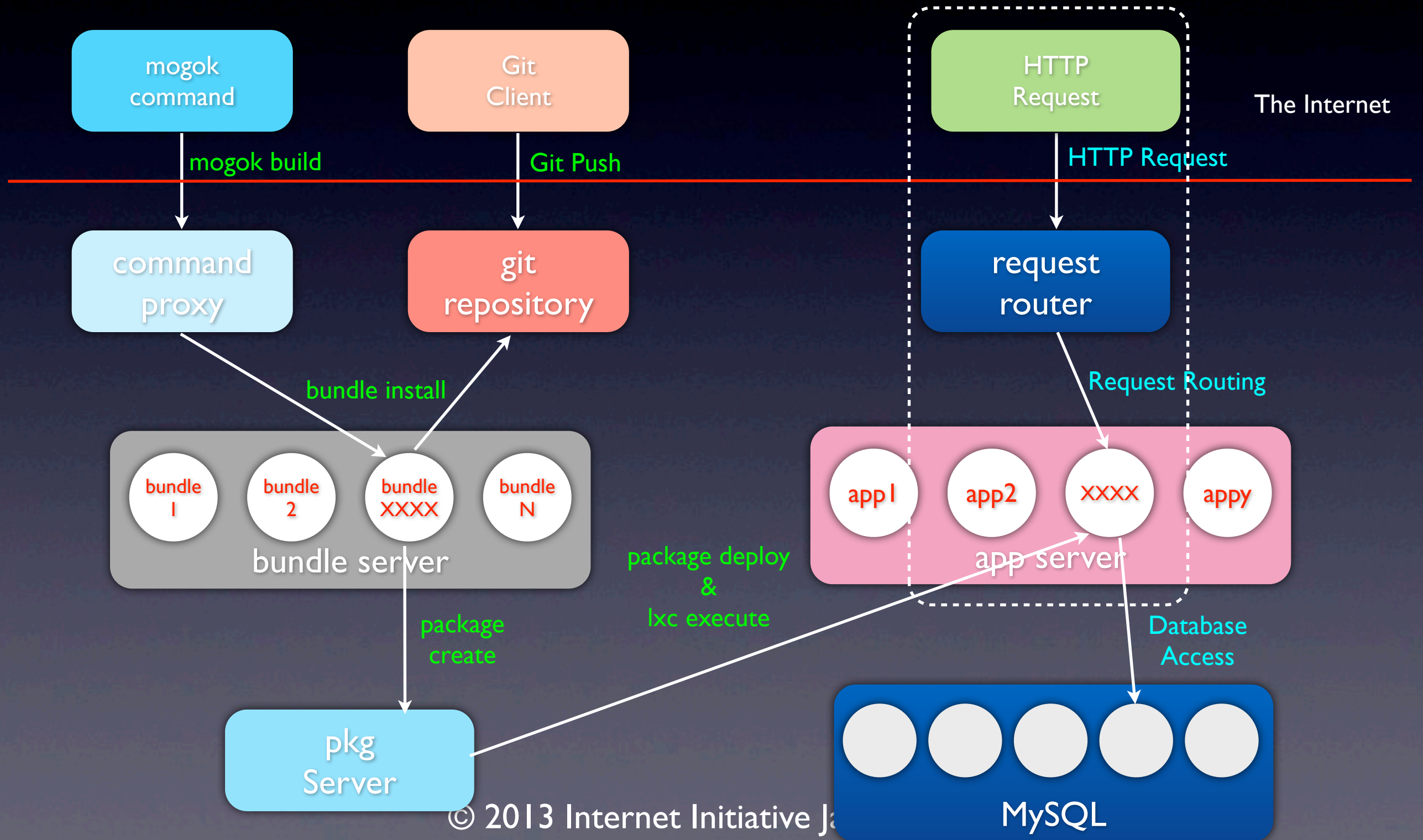


3つのポイント

- リクエストルーティング
- bundle install
- LXC

リクエストルーティング

リクエストルーティング



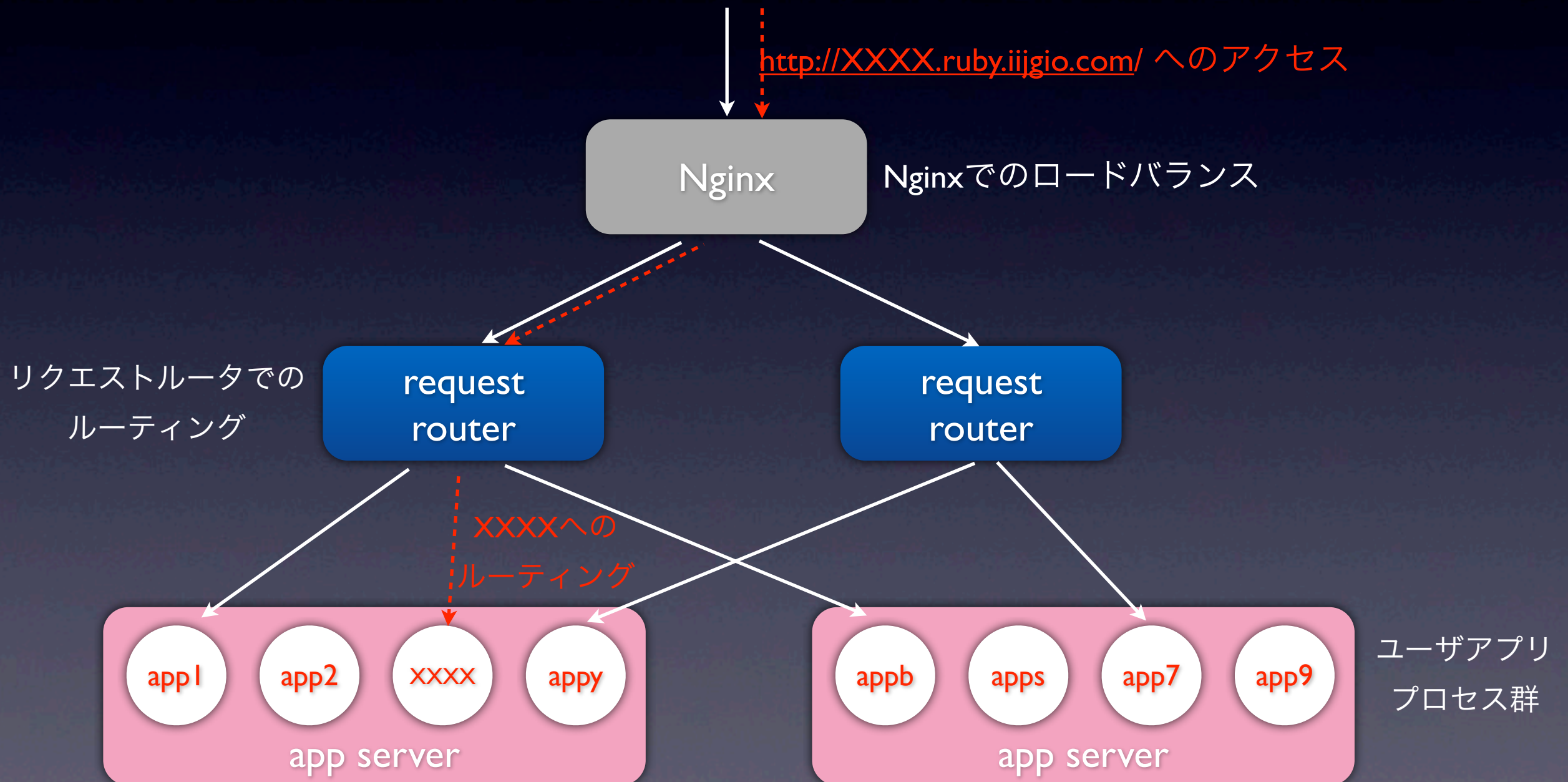
リクエストルーティング

- 外からのHTTP/HTTPSリクエストはまず何で受けているか？
 - Nginx
- Nginxから直にユーザアプリへとリダイレクトされるのか？
 - リクエストをルーティングしないとアプリへ届かない
- リクエストルータを利用
 - <http://XXXXX.ruby.iijgio.com/> のXXXXX解析し、アプリケーションサーバで動作するユーザアプリへとリクエストを転送する

リクエストルーティング

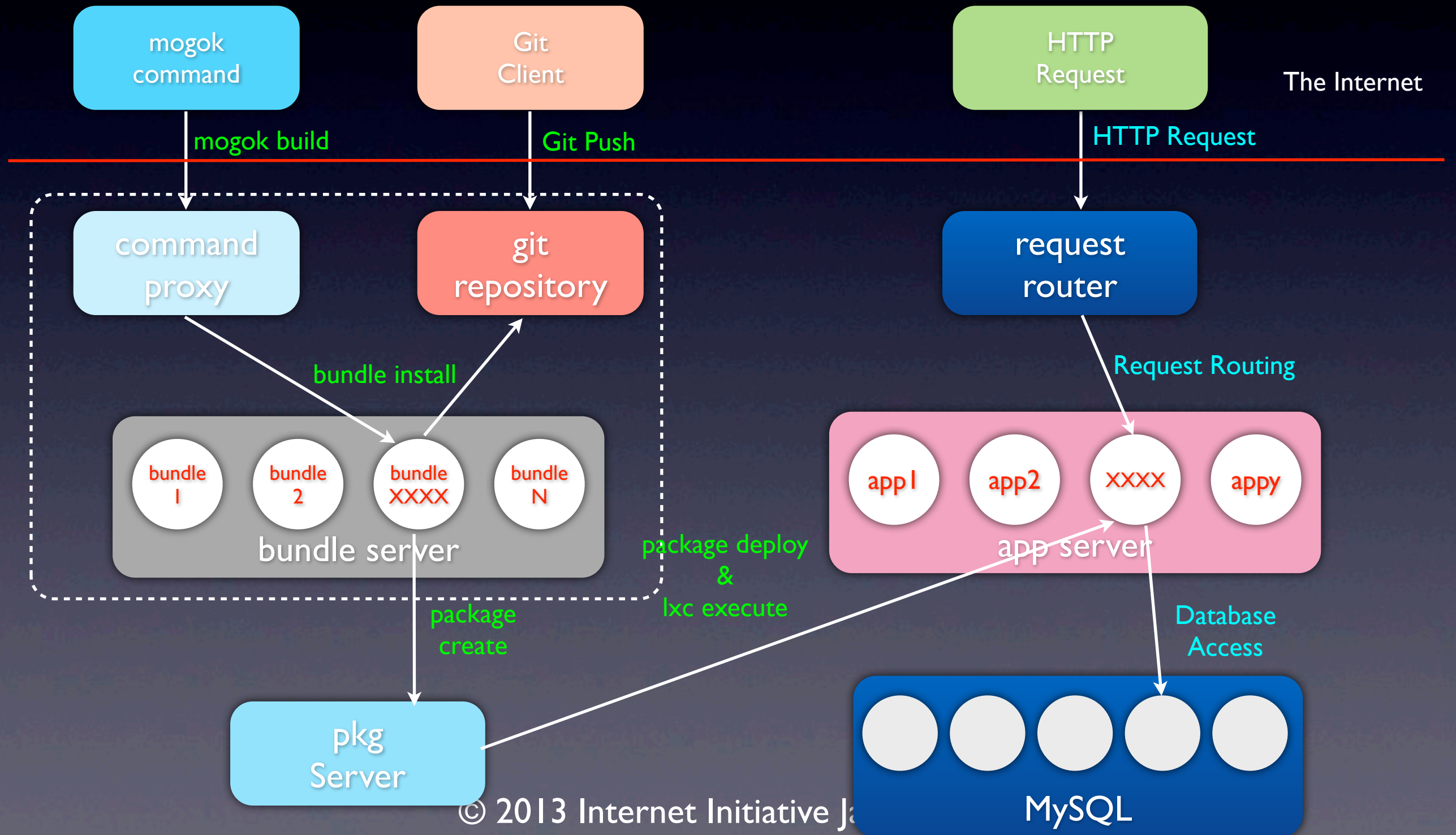
- リクエストルータ
 - Nginxから受け取ったHTTP RequestのURLを解析して、正しいアプリへとリクエストを転送する
 - ルーティング情報はキャッシュされる
 - 新しいルーティング情報が追加・削除されるたびに情報は注入される(ユーザが新しいアプリをデプロイ/削除した時など)

リクエストルーティング



bundle install

bundle install



bundle installとは？

- Gem
 - Rubyで使われるアプリやライブラリのパッケージ
- RubyGems
 - Gemを管理するシステム
 - `gem install XXX` で必要なアプリをインストールできる
 - rubygems.orgでたくさんのライブラリが公開されている
- Bundler
 - BundlerはRubyGemsと一緒に使われるアプリケーションの依存関係を管理するツール
 - `bundle install`を実行すると必要なライブラリがインストールされる

MOGOKでbundle install

- mogokコマンドにbundle installは存在しない？
- 「mogok build」を実行するとサーバサイドでbundle installが自動的に実行される
 - Gemfileにインストールしたいライブラリを明記
 - Gemfileはgit pushしておく

```
source 'https://rubygems.org'
```

```
# Bundle edge Rails instead: gem 'rails',  
github: 'rails/rails'  
gem 'rails', '4.0.0'
```

```
# Use sqlite3 as the database for Active  
Record  
gem 'sqlite3'
```

Gemfile例

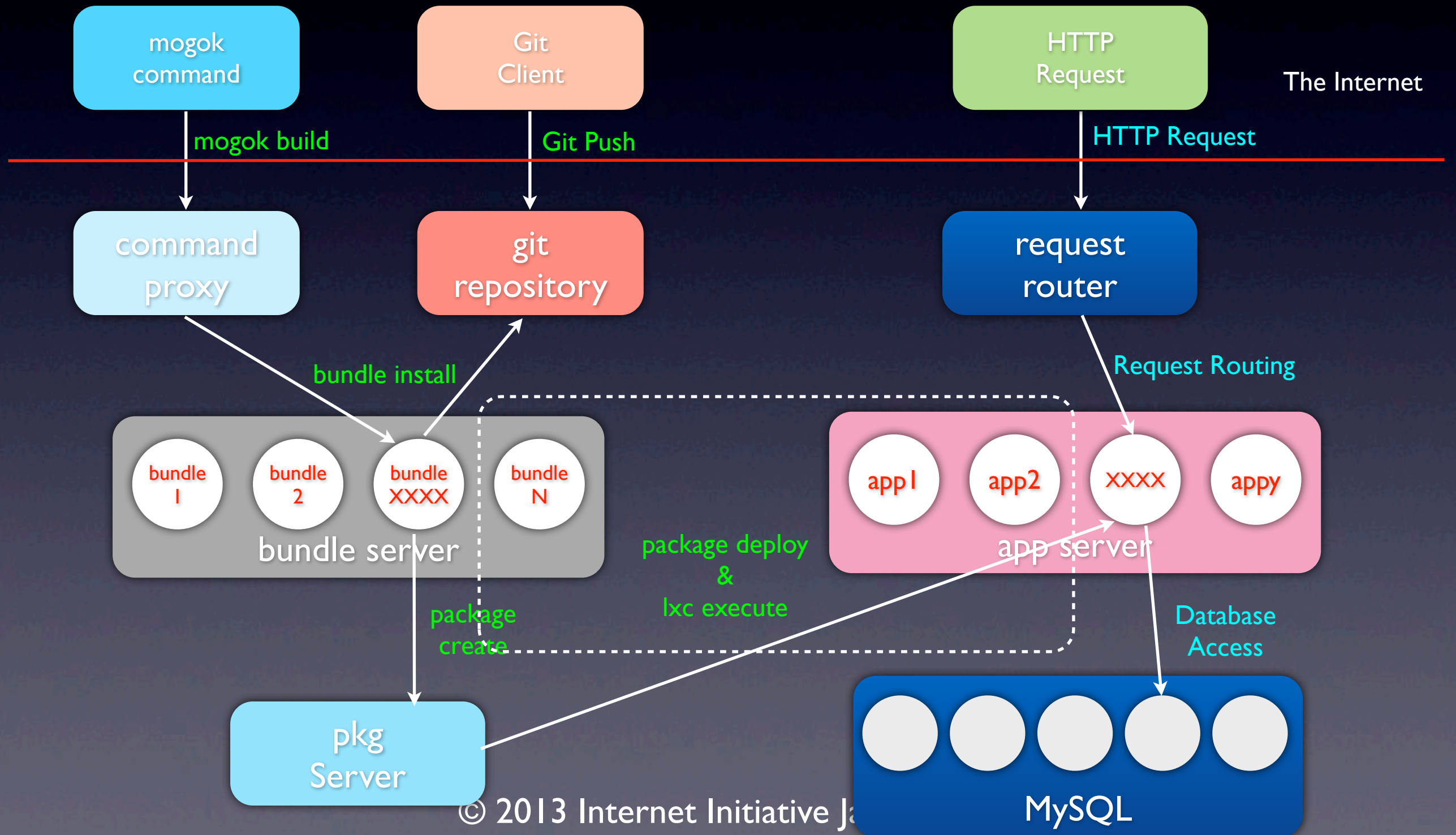
mogok buildコマンド

1. gitリポジトリからソースコード取得
2. bundle installの実行
3. 実行アプリケーションのパッケージ作成
 - Gemfileに指定されたライブラリを含む
 - 「mogok start」時に作成されたパッケージがアプリケーションサーバで実行される

buildの自動化をMOGOKシステム側で行う

LXC

LXC



LXCで何を実現したいか？

- ユーザのアプリの隔離
 - 他のユーザへと悪さをさせない
 - 動作するサーバ（母艦）への悪さをさせない
- `bundle install`実行時に悪意のある挙動を防ぐ
 - gitサーバにソースコードを保存できるので、悪意のあるプログラムを保存／実行可能
 - 外部のプログラムも安全である保証はない

LXCについて

lxc Linux Containers
container namespace cgroup virtualisation

- Linux Container
- 完全仮想化ではなく一部リソースの共有
- cgroupとnamespace
 - リソース管理、リソース隔離

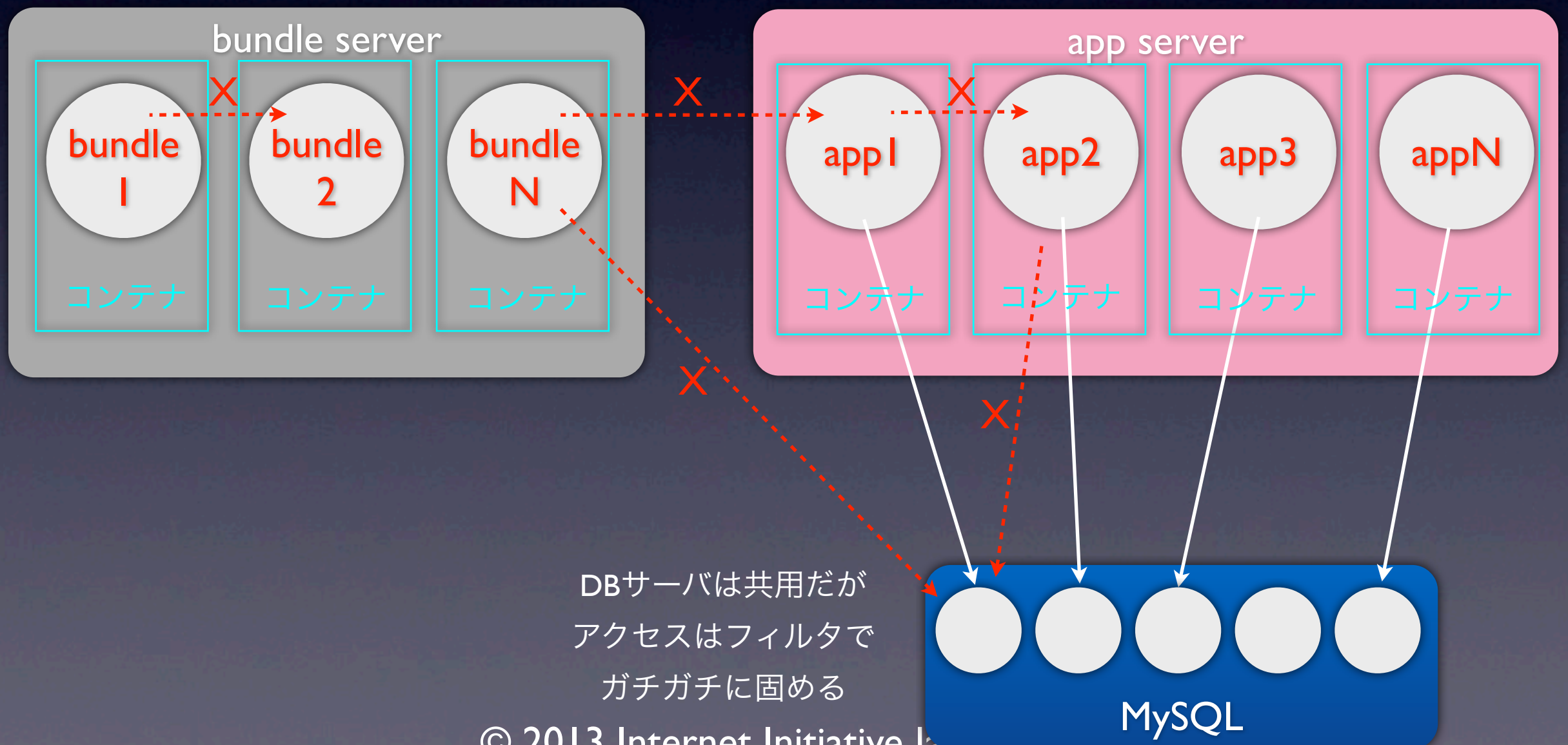
LXCの2種類のコンテナ

- LXCには2種類のコンテナが存在する
 - アプリケーションコンテナ（特定のプロセスをコンテナ内で実行）
 - システムコンテナ（独立してOSをコンテナ内で実行、VMと同等のイメージ）
- MOGOKではアプリケーションコンテナを採用
 - Ruby on Railsを動作させるための専用コンテナイメージを作成

LXCを利用した動作環境

bundle installは個々のLXCコンテナで行われる。コンテナ間の通信は禁止。

ユーザアプリは個々のLXCコンテナで実行される。コンテナ間の通信は禁止。



MOGOKの今後

MOGOKの今後

- 有償版
- 無償版

有償版

- 法人向けサービス
- サービススペック
 - 複数アプリケーション（最大20アプリ）
 - 複数インスタンス対応（アプリあたり5インスタンスまで）
 - Worker（非同期実行ジョブ）
 - Job（cronのような物）
 - 専用DBサーバ
 - DB容量20GB（オプションで300GBまで増加可）

無償版



- 無償版も継続
 - 1 Webインスタンスを無料で提供
- 有償版は法人向けだから個人で契約できない？
 - 今の所できないが救済処置あり
 - 使い込んでくれた方 / 宣伝してくれた方には無料（契約無し）で有償機能を利用していただけるとのようになりました :-)

まとめ

- IIJのPaaSについて考え方とMOGOKについて紹介しました
- PaaSの本質とMOGOKのシステム説明、MOGOKの今後について解説しました

Q&A