

openstack

Open source software to build public and private clouds.

ITを変えるクラウド基盤技術の最前線

2013.10.4

日本OpenStackユーザ会 中島倫明

- はじめに
 - 日本OpenStackユーザ会の紹介
 - 【復習】クラウドとは
- OpenStackの概要
- 利用状況
- はじめ方
- まとめ

はじめに ユーザ会の紹介

- 国内でのOpenStackの普及活動・人材育成を目的に2010年に発足
 - 運営団体数 19社（学術機関、任意団体を含む）
 - 会員数 982名（2013/10/1時点）



The screenshot shows the homepage of the Japan OpenStack User Group (JOSUG). The header includes the group's name in Japanese and English, along with the URL 'openstack.jp'. A navigation menu contains links for TOP, Community, News, Press Release, Documents, Wiki, Events, UserStories, and Links. A sidebar menu on the left lists these same categories. The main content area features a title '日本OpenStackユーザ会' and a sub-header '日本OpenStackユーザ会 (JOSUG)について'. Below this, there is a welcome message in Japanese and a paragraph of introductory text. The OpenStack JAPAN logo is displayed at the bottom of the main content area.

■ 主な活動

- メーリングリストでの情報交換
- イベント開催、勉強会、セミナーの開催（関東中心）
- オープンソースカンファレンス出展（関東、関西、東海）
- 翻訳活動



■ イベントのご紹介



OpenStack Days Tokyo 2014

広がるオープンクラウド エコシステム

会期: 2014年2月13日(木)、14日(金); 2日間開催

会場: ソラシティカンファレンスセンター (千代田区)

基調講演

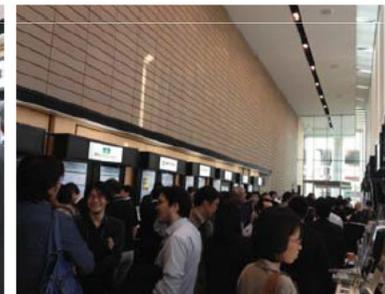
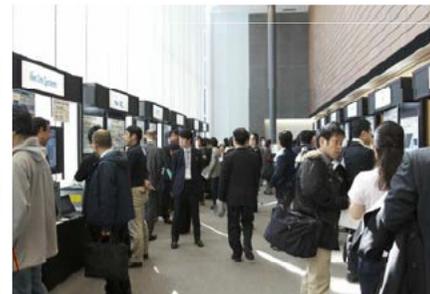
- (1) 本家USコミュニティからの招待講演
- (2) 国内大規模ユーザ事例

スポンサーセッション

国内外の企業から最新技術・製品・事例紹介など多数

■ 前回 OpenStack Day Tokyo 2013

事前登録者数	当日来場者数	スポンサー企業数	メディア協力媒体数
1038	700+	22	13



OpenStack Summit 香港



■ 2013/11/5~8



[Home](#) [Software](#) [User Stories](#) [Community](#) [Foundation](#) [Profile](#) [Blog](#) [Wiki](#) [Documentation](#)



// DEVELOPERS // USERS // OPS

香港 HONG KONG

NOVEMBER

5	6	7	8
T	W	T	F

 2013

The OpenStack Summit is a four-day conference for developers, users, and administrators of OpenStack Cloud Software.

Register 注册 >

グローバルでの事例紹介
次期OpenStack開発の方向性の議論

はじめに

【復習】クラウドとは

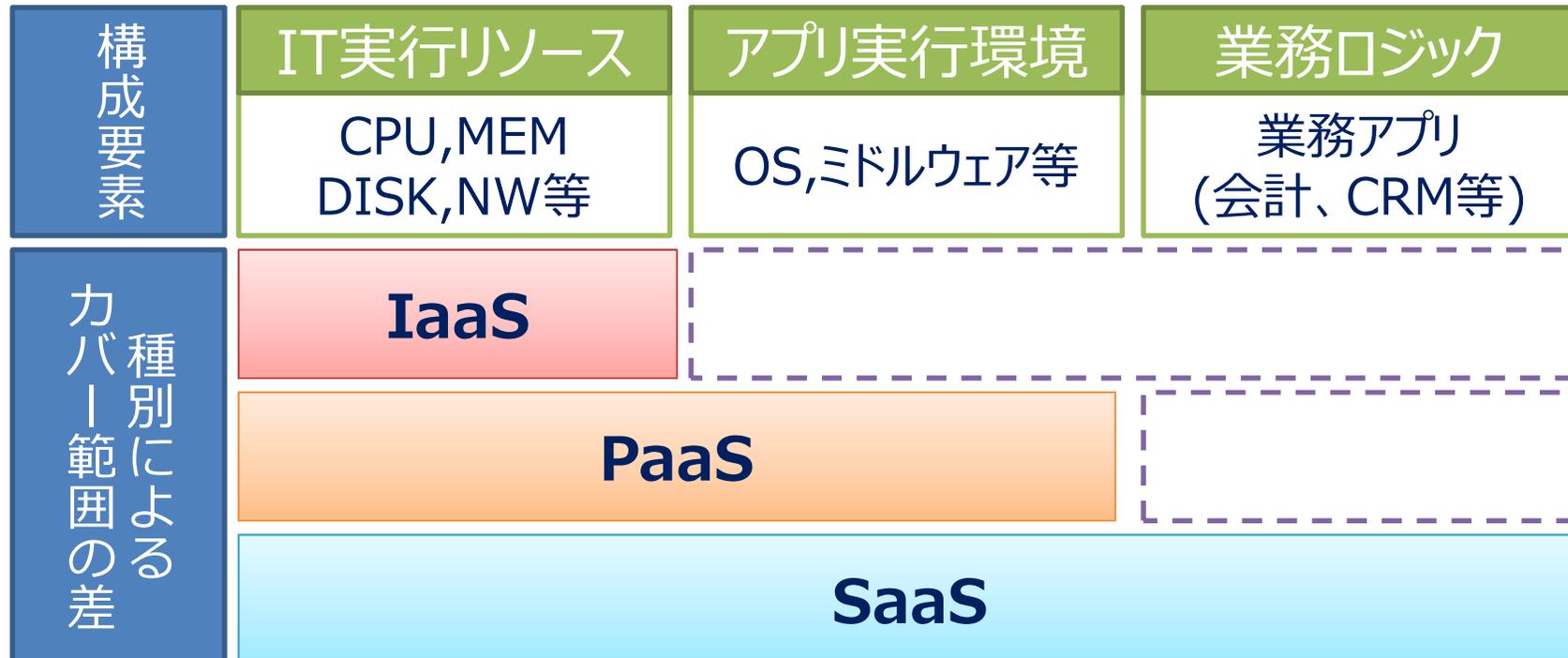
利用者からみたクラウド



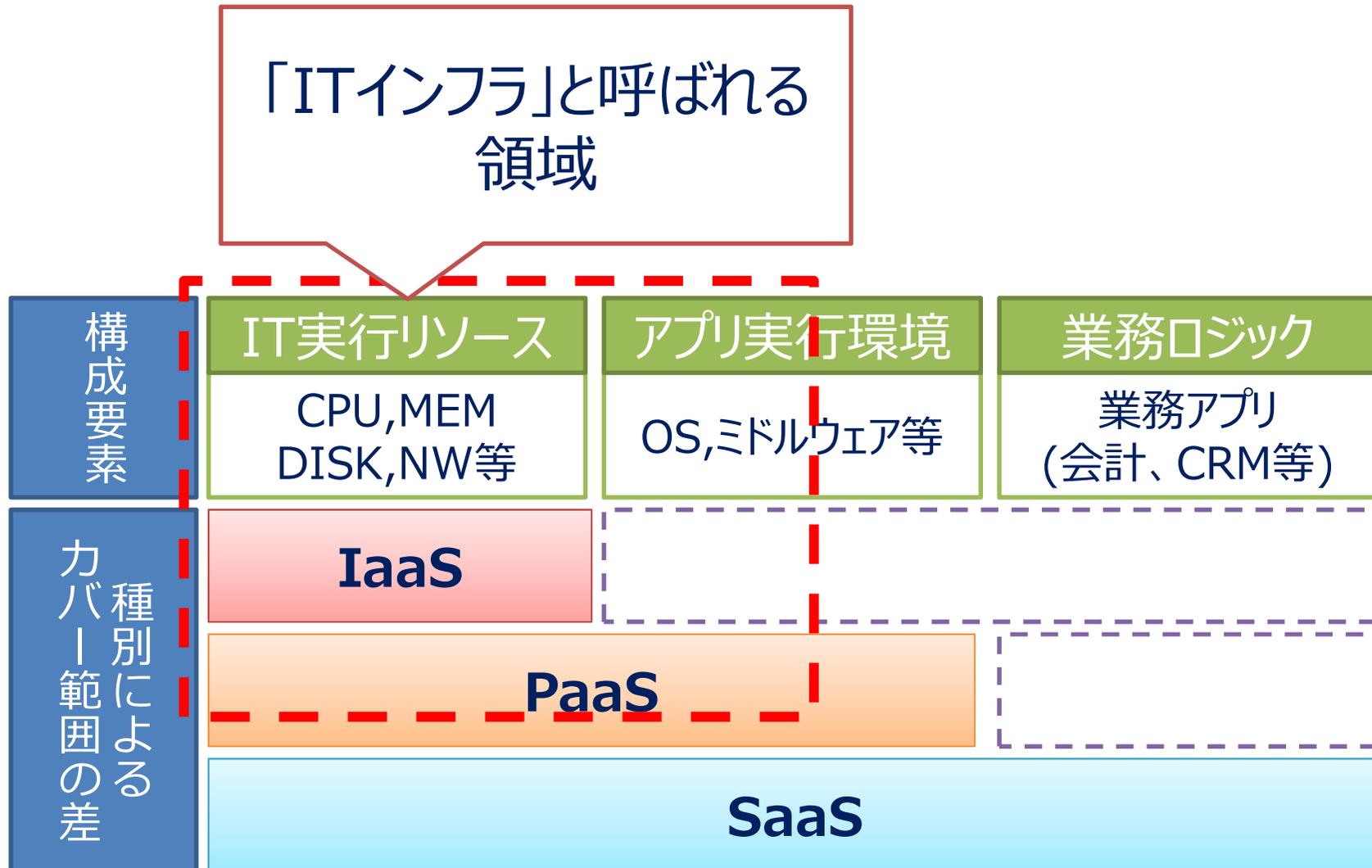
■一言で言うと・・・

- ITシステムを所有せず、従量課金で利用する方式です。

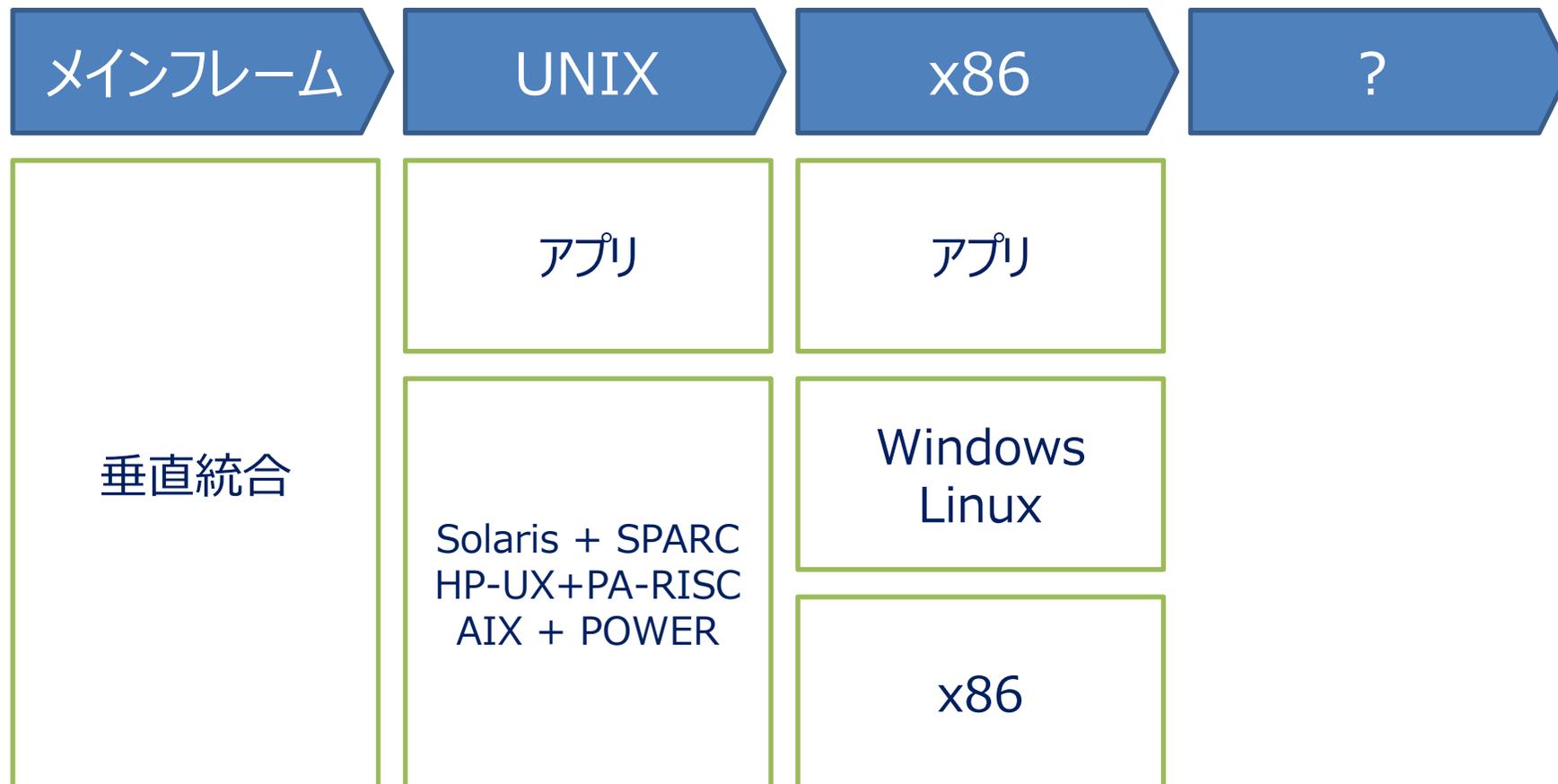
■クラウドの大まかな種別



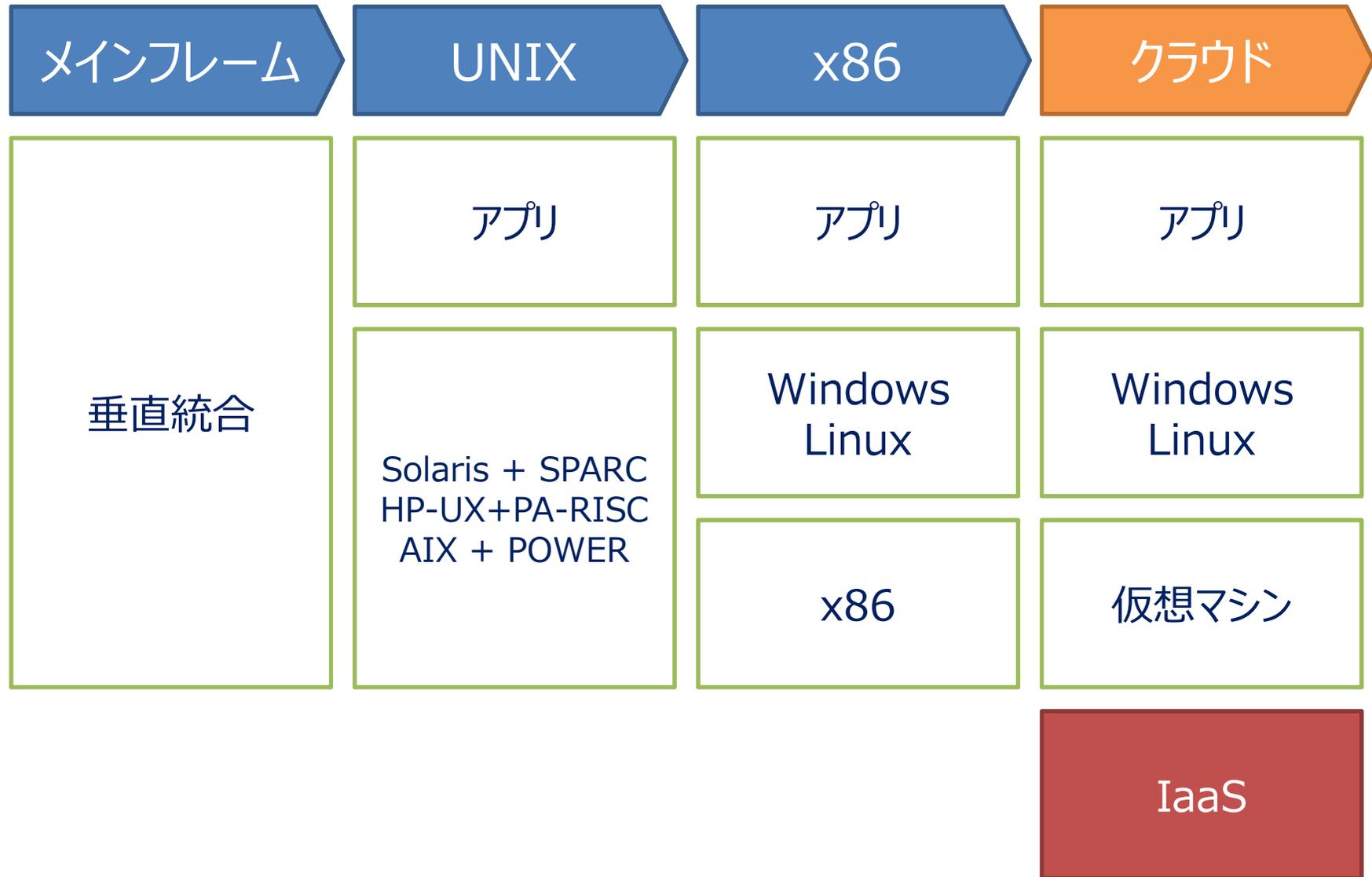
本日の話題



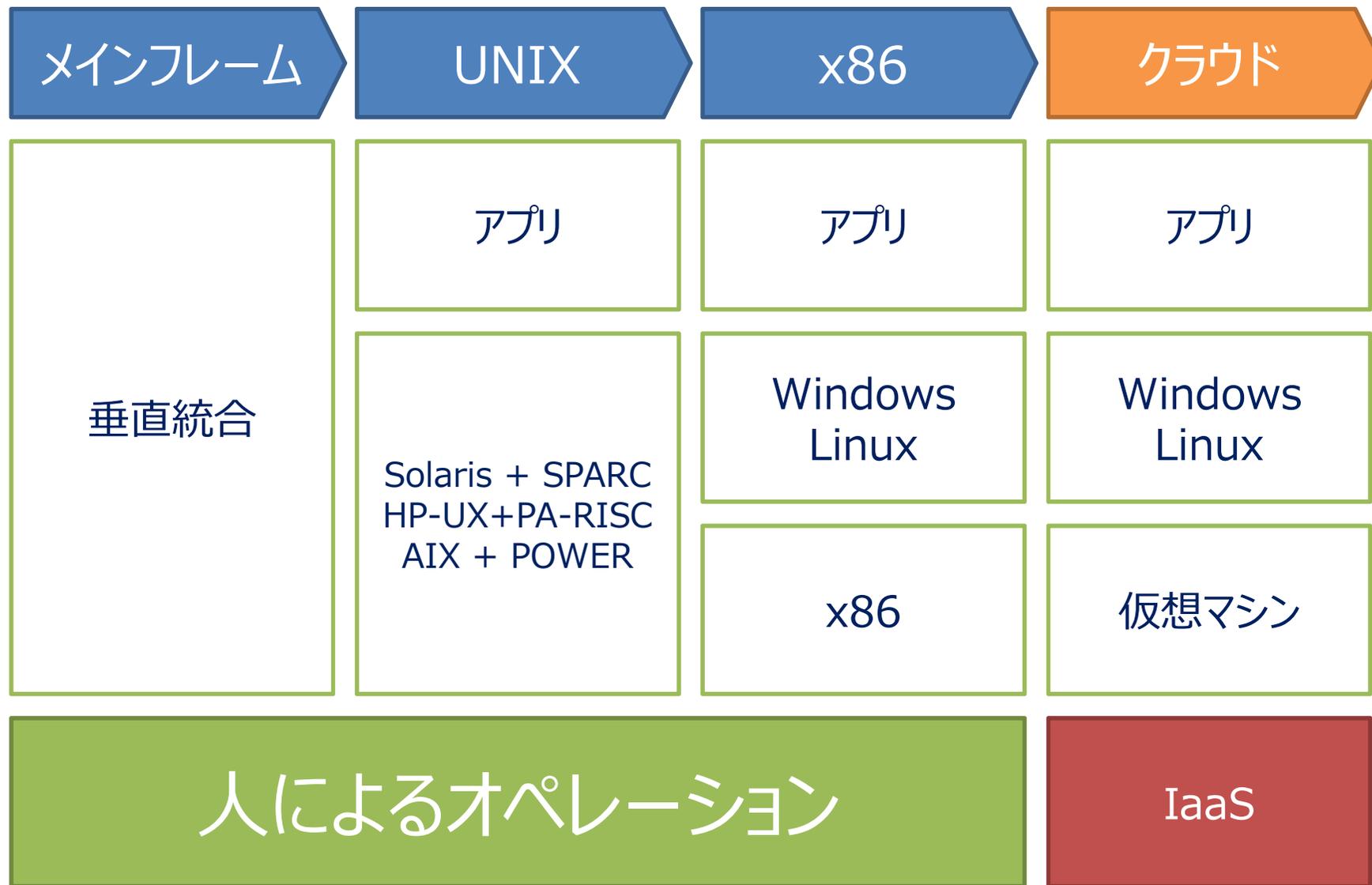
ITインフラの技術変遷



クラウド時代のアーキテクチャ



IaaSが変えたもの



IaaSが生み出すメリット



Before

サーバの構築は人が行う

ネットワーク接続は人が行う

増設や変更は人が行う

After

IaaSが生み出すメリット



Before

サーバの構築は人が行う

ネットワーク接続は人が行う

増設や変更は人が行う



規模が大きいと人手も増加

繰り返し作業でも、
N回分の工数が必要

After

IaaSが生み出すメリット



Before

サーバの構築は人が行う

ネットワーク接続は人が行う

増設や変更は人が行う



規模が大きいと人手も増加

繰り返し作業でも、
N回分の工数が必要



やりたい事がITインフラに制限される

After

IaaSが生み出すメリット



Before

サーバの構築は人が行う

ネットワーク接続は人が行う

増設や変更は人が行う



規模が大きいと人手も増加

繰り返し作業でも、
N回分の工数が必要



やりたい事がITインフラに制限される

After

サーバの構築はプログラムが行う

ネットワーク接続はプログラムが行う

増設や変更はプログラムが行う

IaaSが生み出すメリット



Before

サーバの構築は人が行う

ネットワーク接続は人が行う

増設や変更は人が行う



規模が大きいと人手も増加

繰り返し作業でも、
N回分の工数が必要



やりたい事がITインフラに制限される

After

サーバの構築はプログラムが行う

ネットワーク接続はプログラムが行う

増設や変更はプログラムが行う



規模が大きくても変わらない

何度でも同じ操作が
1回分の工数で可能

IaaSが生み出すメリット



Before

サーバの構築は人が行う

ネットワーク接続は人が行う

増設や変更は人が行う



規模が大きいと人手も増加

繰り返し作業でも、
N回分の工数が必要



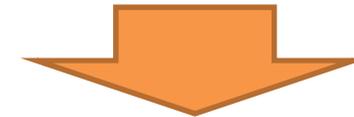
やりたい事がITインフラに制限される

After

サーバの構築はプログラムが行う

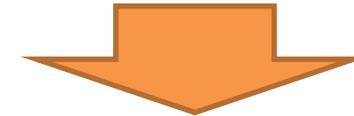
ネットワーク接続はプログラムが行う

増設や変更はプログラムが行う



規模が大きくても変わらない

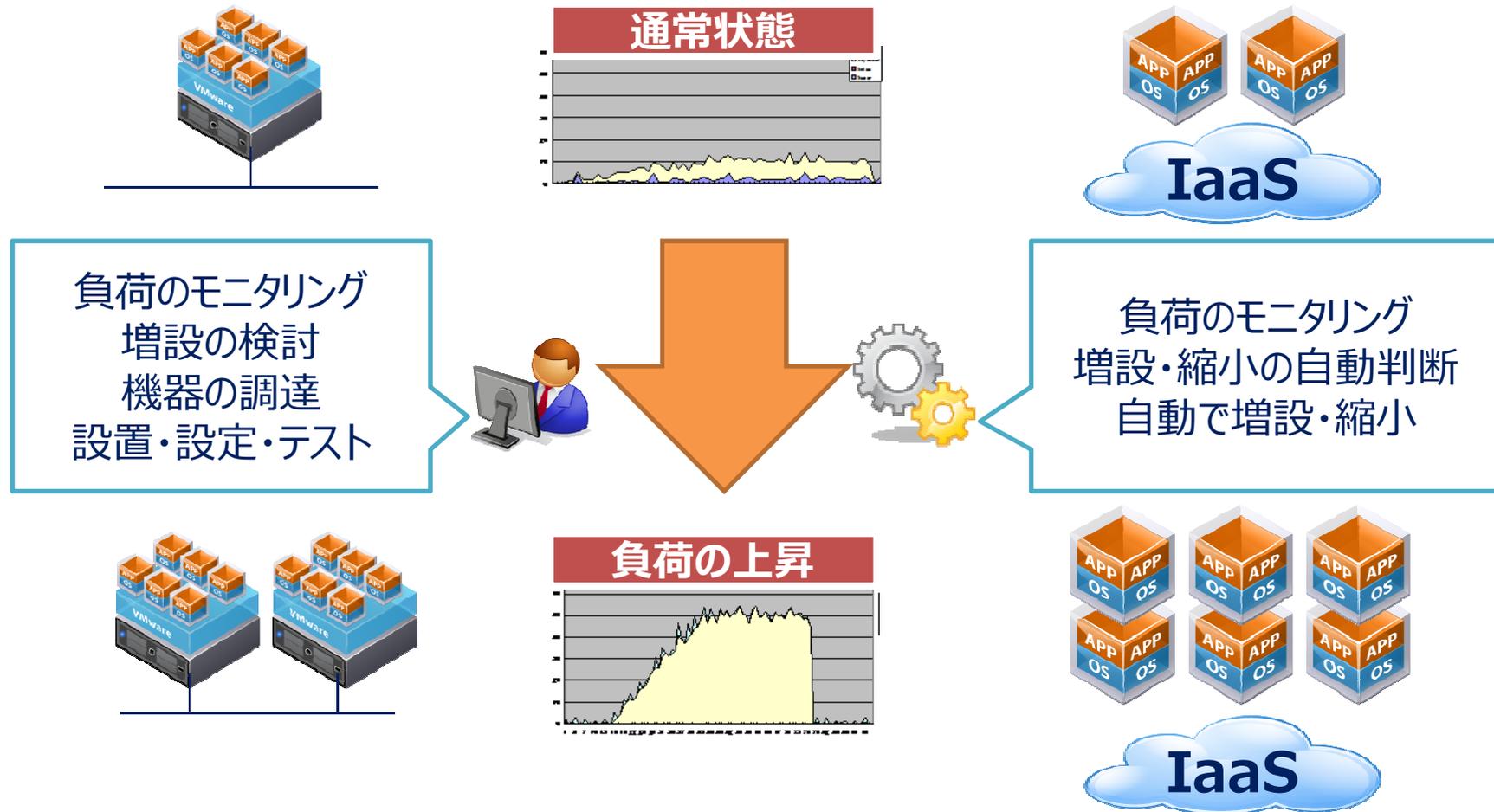
何度でも同じ操作が
1回分の工数で可能



やりたい事に合わせて
ITインフラをプログラムで制御できる

典型的な活用例

■ サービスの負荷状態に応じたオートスケール



- 単なる仮想リソース貸しではなく、

プログラムから制御可能なITインフラ環境

IaaSを活用するには



■ 既にあるサービスを利用する

- 様々なパブリックサービスが既に存在します。

IaaSを活用するには



■ 既にあるサービスを利用する

- 様々なパブリックサービスが既に存在します。

■ 自分で作る

- プライベート、コミュニティクラウドを構築する
- 自分でパブリックサービスを立ち上げる

IaaSを活用するには

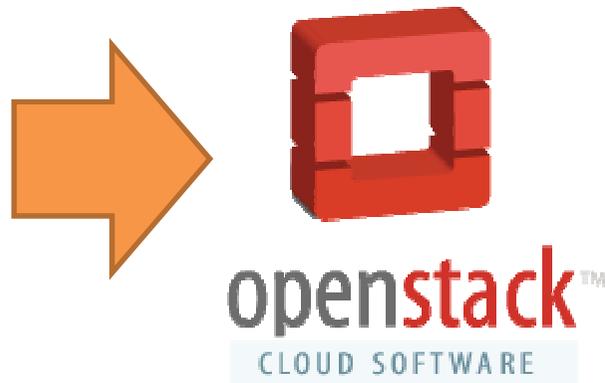


■ 既にあるサービスを利用する

- 様々なパブリックサービスが既に存在します。

■ 自分で作る

- プライベート、コミュニティクラウドを構築する
- 自分でパブリックサービスを立ち上げる



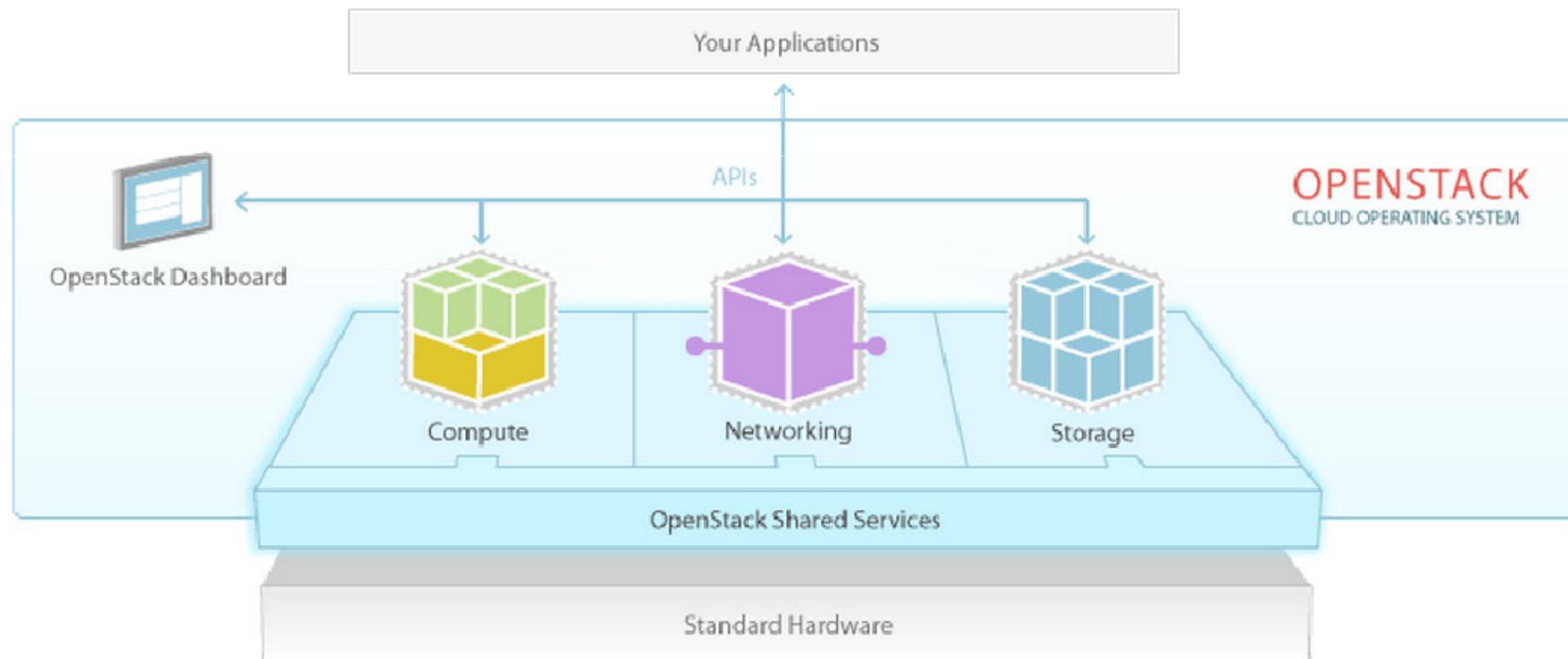
類似するソフトウェア



OPENSTACKとは

■ OSSで開発されるCloudOS

OpenStack: The Open Source Cloud Operating System



ソフトウェアの特徴



OpenSource

▶ **オープンソース**のクラウド基盤ソフトウェア
Apache2ライセンス

World Wide

▶ 世界各国から多数の
企業・開発者が参加

Community

Join our global community of technologists, developers, researchers, corporations and cloud computing experts.

12211

PEOPLE

130

COUNTRIES

IaaS

▶ 仮想サーバ、ストレージ、仮想NWをはじめ、
様々な機能を搭載(一部PaaSの機能も)

Standardization

▶ 物理環境を隠蔽し、**標準化された**ITインフラ

プロジェクトの特徴



- 多数のスポンサー企業による強固な運営基盤と、Foundationによる中立維持

Platinum Members



AT&T



Canonical



HP



IBM



Nebula



Rackspace



Red Hat, Inc.



SUSE

- 安定した開発・ユーザーコミュニティの支援体制

実装面での特徴

■ Pythonで実装

■ 豊富な機能

- 仮想マシン、物理マシン
- ネットワーク (L2, L3, L7, セキュリティ)
- ブロックストレージ、オブジェクトストレージ
- イメージ管理、スナップショット管理
- マルチテナント 等々

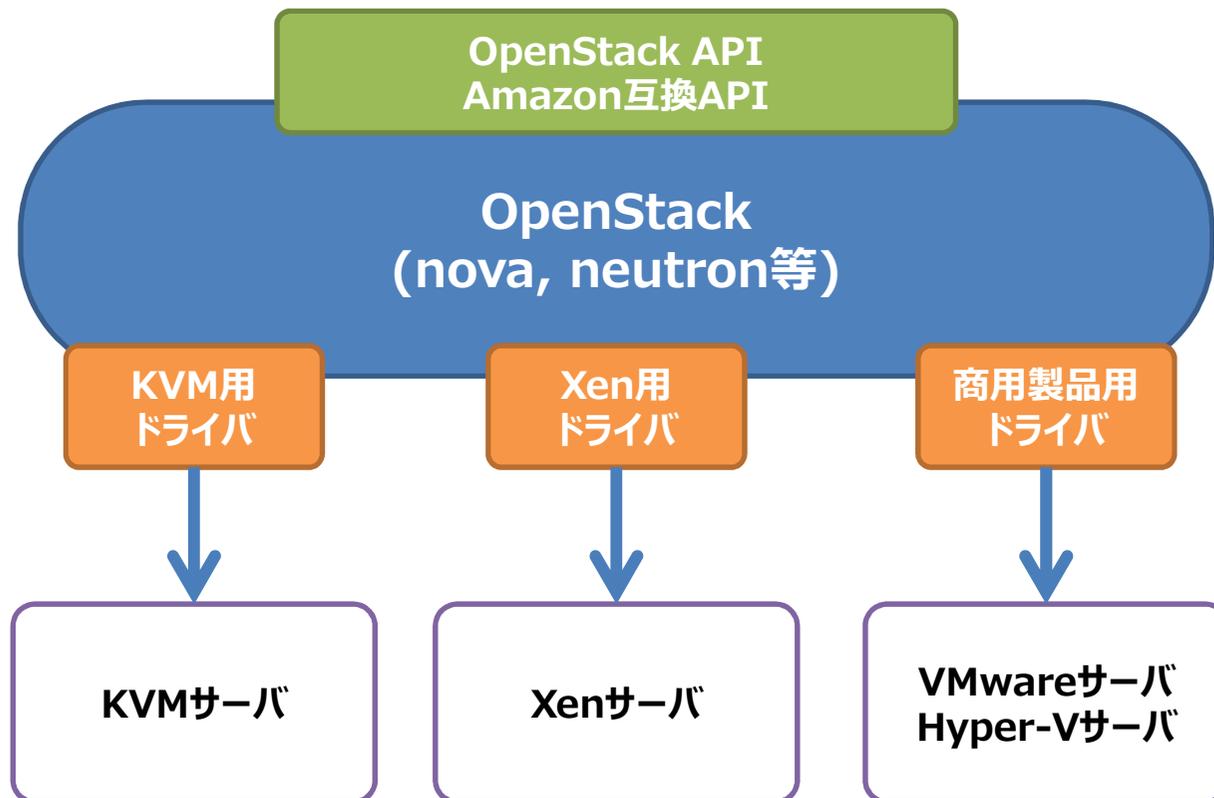
■ マイクロコントローラー構造

- 機能単位でコンポーネントが分離し、独立動作

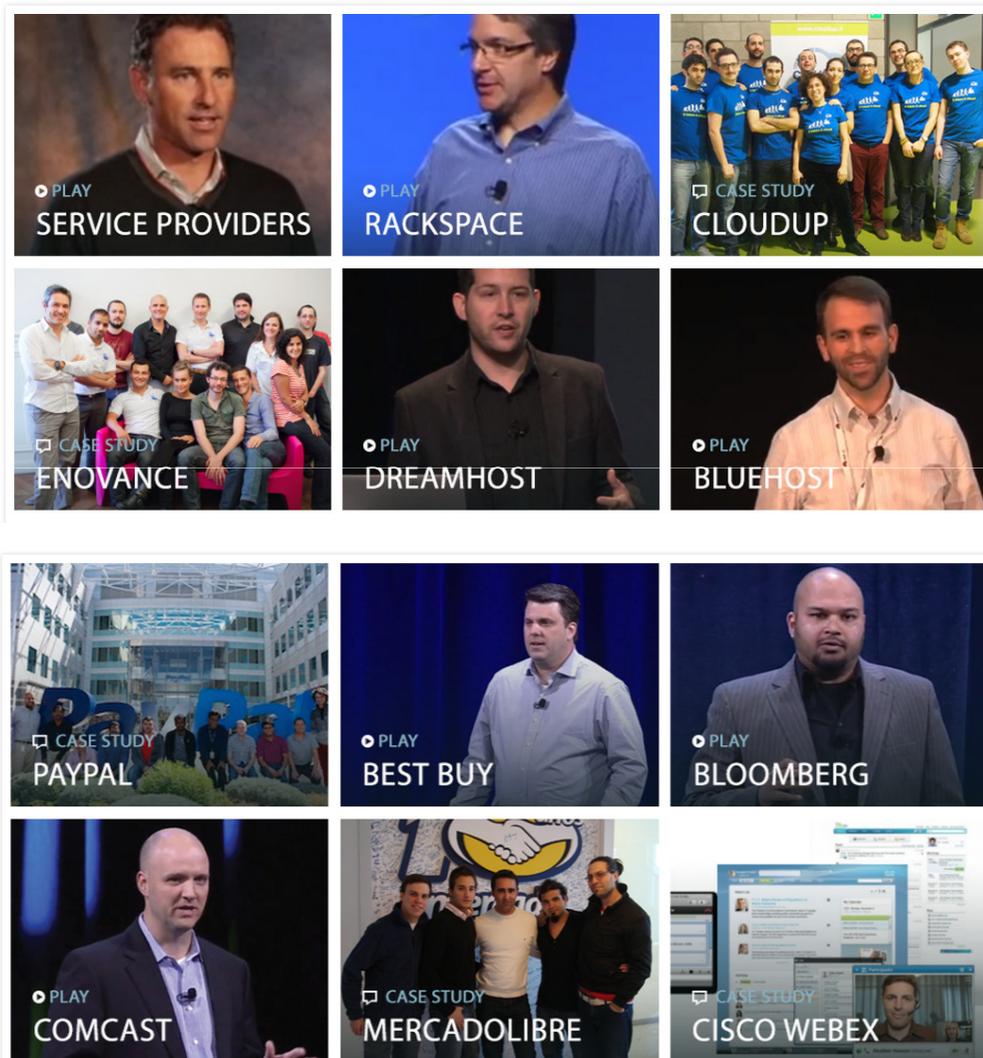
構造的な特徴



ユーザやプログラムは、
OpenStackの利用方法だけを
知っていれば環境の操作が可能



豊富な事例（一部抜粋）



利用業種

- ・Webサービス業
- ・SaaSプロバイダー
- ・eコマース
- ・学術
- ・研究開発機関
- ・政府機関
- ・情報産業
- ・ヘルスケア
- ・クラウドプロバイダー
- ・MSP
- ・通信産業
- ・ゲーム
- ・映像
- ・メディア

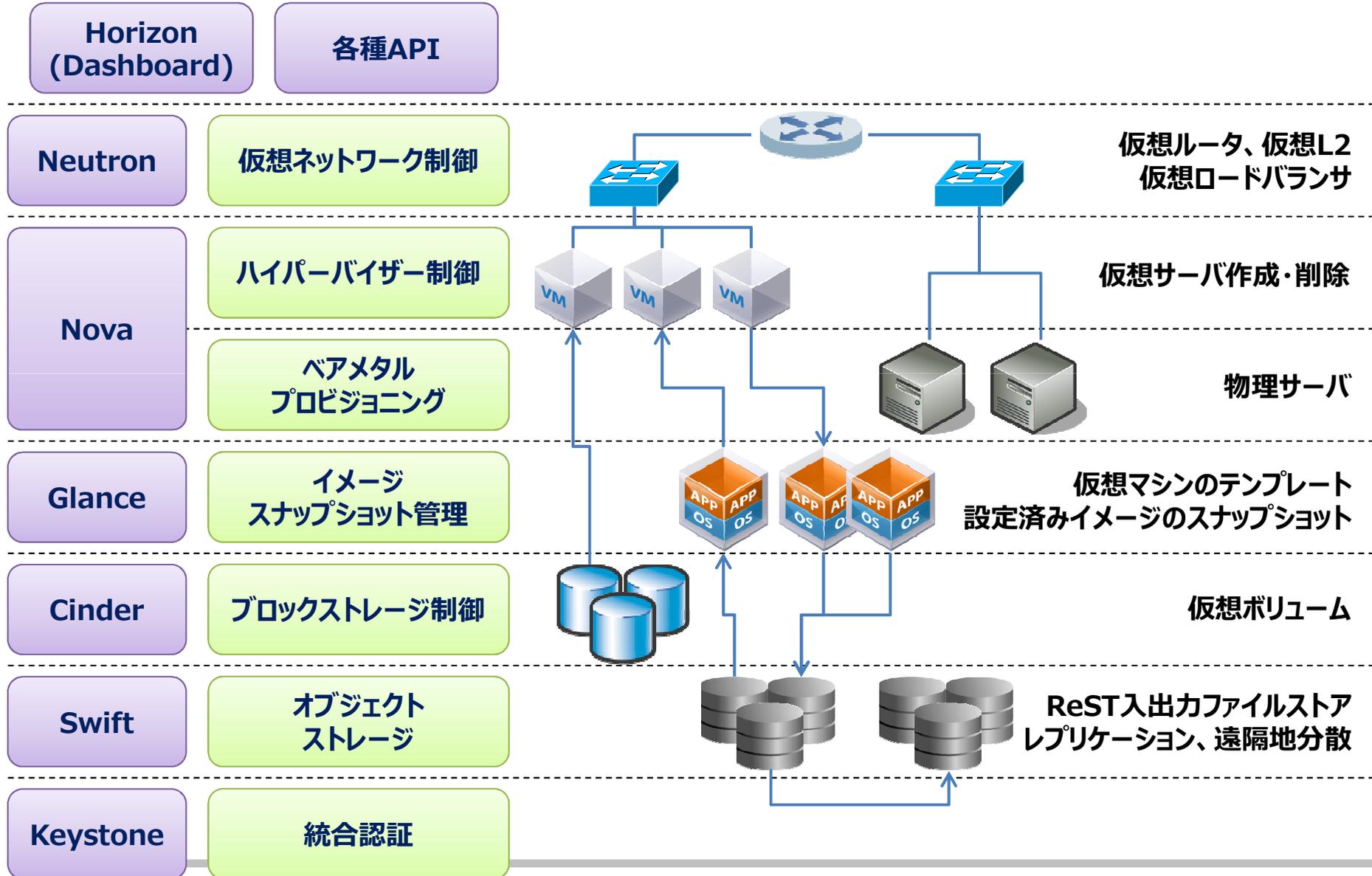
継続的なアップデート



version	description
Austin	Nova/Swift 初期リリース
Bexar	Glance実装。Nova/Swift連携可能に。
Cactus	VMware, LXC対応等の機能強化
Diablo	スケジューラー、レプリケーション等の機能追加
Essex	Keystone/Horizon追加。
Folsom	Neutron(当時はQuantum)/Cinder追加
Grizzly	対応HyperVisor, Storageの追加、機能強化



機能イメージ



操作イメージ

Launch Instance

Details | Access & Security | Networking | Volume Options | Post-Creation

Instance Source
Image

Image
cirros-0.3.0-x86_64-uec

Instance Name
hadoopvm

Flavor
m1.tiny

Instance Count
100

インスタンスを起動するために詳細を指定します。
以下の図は、このプロジェクトにより使用されているリソースをプロジェクトのクォータと関連付けて表示します。

インスタンスタイプの詳細

名前	m1.tiny
仮想 CPU	1
ルートディスク	0
一時ディスク	0
合計ディスク	0
RAM	512 MB

プロジェクトのクォータ

インスタンス数 (0)	0
仮想 CPU 数 (0)	0
Total RAM (0 MB)	0

セキュリティグループのルールの変更

Delete Rules

セキュリティグループ

<input type="checkbox"/>	IP プロトコル	ポート番号 (下限)	ポート番号 (上限)	接続元	アクション
<input type="checkbox"/>	TCP	22	22	0.0.0.0/0 (CIDR)	ルールの削除します
<input type="checkbox"/>	TCP	80	80	0.0.0.0/0 (CIDR)	ルールの削除します
<input type="checkbox"/>	TCP	443	443	0.0.0.0/0 (CIDR)	ルールの削除します

Displaying 3 items

ルールの追加

IP プロトコル	ポート番号 (下限)	ポート番号 (上限)	元グループ	CIDR
TCP			CIDR	0.0.0.0/0

キャンセル | ルールの追加

操作イメージ





DASHBOARD

ネットワークポロジ

次の役割でログイン中: user01 設定 ヘルプ ログアウト OSAKA

インスタンスの起動 ネットワークの作成 ルーターの作成

プロジェクト

現在のプロジェクト
demoTenant1

コンピュートの管理

- 概要
- インスタンス
- ボリューム
- イメージとスナップショット
- アクセスとセキュリティ

ネットワークの管理

- ネットワーク
- ルーター
- ネットワークポロジ**

オブジェクトストア

- コンテナ

The diagram illustrates a network topology with three main network segments:

- ext-network (external):** A green vertical line on the left.
- pub-nw (public network):** A cyan vertical line in the middle, with IP address 172.26.10.0/24.
- internal-nw (internal network):** A purple vertical line on the right, with IP address 172.26.20.0/24.

Key components and connections:

- route1 router:** A black box at the top left, connected to the ext-network and pub-nw. IP: 172.26.10.1.
- Front servers (pub-nw):** Three black boxes connected to the pub-nw:
 - front--4911... server (IP: 172.26.10.4)
 - front--2007... server (IP: 172.26.10.5)
 - front--7863f... server (IP: 172.26.10.2)
- Backend servers (internal-nw):** Two black boxes connected to the internal-nw:
 - backend-0... server (IP: 172.26.20.7)
 - backend-0... server (IP: 172.26.20.6)
- Inter-network connections:** Lines connect front servers to the internal-nw:
 - front--4911... server to internal-nw (IP: 172.26.20.3)
 - front--2007... server to internal-nw (IP: 172.26.20.4)
 - front--7863f... server to internal-nw (IP: 172.26.20.2)

```
#!/usr/bin/env python

from novaclient.v1_1 import client
nt = client.Client("demo",
                  "openstack",
                  "demo",
                  "http://11.22.33.44:5000/v2.0",
                  service_type="compute")

ns = nt.servers.create(name = "testvm001",
                       image = im[0],
                       flavor = fv[0],
                       nic = nw[1])
```

利用状況

3つのフェーズ



– Phase1

- Amazonライクなサービスを始めたいクラウド事業者が利用を開始。または研究・開発用の環境。

– Phase2

- OpenStackを標準化のツールとして活用し、自社のサービスやITシステムの動作環境として利用。

– Phase3

- 標準化が進行し、専用ソフトウェア(IaaS上でネイティブに動作する)が登場することで、プラットフォームとして定着。

現在の状態



- 現在はPhase1の「IaaS基盤」の構築が最も活発
- 一部の先進ユーザが自社環境の標準化のためにPhase2への取り組みを行っている。
- Phase3はまだ研究開発の段階

はじめ方

■ OpenStackは難しい！

– かつてはそうでしたが、今は簡単

- かつて

- インストールそのものが既に難解
- ドキュメントが間違っている（全く無い）
- 動きが説明と違う

– メジャーLinuxディストリビューションには標準搭載

– KVMやVirtualBox上の仮想マシンでも動作可能

- ただし動作検証や開発での利用になります。

豊富な構築支援ツール



ツール名	OS自動構築		OpenStack自動構築			GUI	備考
	Red Hat 対応	Ubuntu 対応	Red Hat 対応	Ubuntu 対応	複数台環境 対応		
Crowbar	×	○	×	○	○	○	・ Chefベース ・ DELL開発
Cobbler	○	○	—	—	—	○	・ OS自動構築のみ
Spiceweasel	—	—	?	○	○	×	・ Chefベース
PackStack	—	—	○	×	○	×	・ Puppetベース
OpenCenter	—	—	○	○	○	○	・ Chefベース ・ RackSpace開発
ansible	—	—	○	○	○	×	・ Python ・ SSHによるPUSH型
devstack	—	—	×	○	×	×	・ シェルスクリプト ・ All in One構成 ・ 開発環境向け
anvil	—	—	○	×	×	×	・ シェルスクリプト ・ All in One構成 ・ 開発環境向け

OpenCenterを使った構築例



A screenshot of the OpenCenter dashboard interface. At the top left, there is a logo for "RACKSPACE™ PRIVATE CLOUD POWERED BY OPENSTACK™". To its right is a button labeled "OpenCenter". Below this is a section titled "Workspace" with a gear icon. Underneath, there are three main panels: "Available Nodes" (empty), "Service Nodes" (containing two nodes labeled "eco1" and "eco2"), and "NovaCluster" (containing two sub-panels: "Infrastructure" with nodes "eco4" and "eco3", and "Compute" with a sub-panel "AZ nova" containing nodes "eco5" and "eco6"). Each node has a green vertical bar to its left and a gear icon to its right.

OpenStackへの入門はもう簡単！



かつてのOpenStack

現在のOpenStack



■ OpenStack Operations Guide

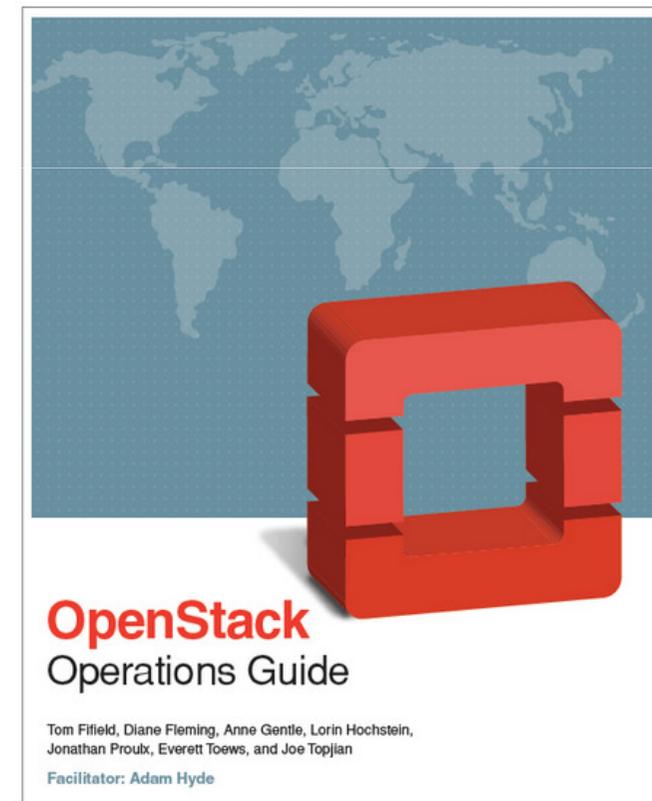
- <http://docs.openstack.org/ops/>

- 設計・構築・運用に係わるノウハウ集
- トラブル事例もあり
- 翻訳版もあります(*1)

■ ユーザ会ドキュメント

- これまでの勉強会資料等(日本語)

- <http://openstack.jp/documents.html>



(*1) <http://openstack-ja.github.io/openstack-manuals/openstack-ops/content/>

■ 設計

- やりたいことと、OpenStack がカバーする範囲のマッチング
 - そもそもOpenStackの採用が最適なのか？
 - 利用する場合、足りない部分をどう開発 or カバーするか？

■ 構築

- 利用用途が多彩で、組合せも無数に存在する。
 - 安定・実績のある組合せを選択

■ 運用

- 進化が早いため、計画的なバージョンアップを検討

今後のアップデート

次期バージョンの概要



■ 新コンポーネントの追加

summary	description
【Ceilometer】 OpenStack Metering/Monitoring	課金のためのリソース利用量を集計。 閾値によるアクション設定が可能。
【Heat】 OpenStack Orchestration	オーケストレーション基盤を提供。 レシピに基づいたプロビジョニング、コンフィグレーション。

■ 既存コンポーネントの強化（一部抜粋）

□ スケーラビリティ

✓ 大規模分散を実現する Cell 機能の強化

- ・ 全コンポーネント対応（GrizzlyではNovaのみ対応）
- ・ Cellスケジューラーの強化

✓ マルチサイトレプリケーションへの対応

□ ネットワーク強化（VPNaaS、FWaaS等の追加、Pluginの充実）

□ 性能向上・改善（マルチスケジューラー、ネットワークエンタイトルメント）



IaaS基盤ソフトウェアとして、さらに強固で利便性の高い環境を実現

まとめ

- OpenStack は IaaS を実現するOSS
 - 広範囲のリソースを抽象化
 - 今はIaaS基盤や標準化ツールとしての使い方が主流
 - 数年後にはアプリケーションの実行環境にも

- 実践するには考慮点も多いので注意
 - 始めるのは簡単、まずは体験してみよう！

- openstack.jp に有益な情報アリ