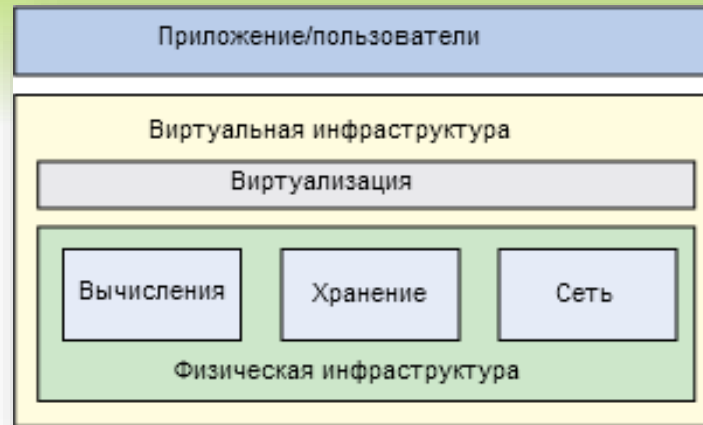


ЛЕКЦИЯ 14: ОБЛАЧНЫЕ ПЛАТФОРМЫ

ЧАСТНЫЕ ОБЛАЧНЫЕ РЕШЕНИЯ

- ① Частные облачные вычисления — это идея IaaS в применении к частной инфраструктуре.
- ① Хотя они и не приносят таких экономических выгод, как общедоступное облако (оплата услуг по факту их использования), но основные принципы облачных вычислений позволяют создать масштабируемую и упругую инфраструктуру внутри корпоративного центра обработки данных.



- ⊙ Архитектуры облачных вычислений, как правило, представляют собой общий набор виртуализированных ресурсов, включая:
 - ⊙ вычислительные ресурсы с различными возможностями,
 - ⊙ ресурсы постоянного хранения данных
 - ⊙ конфигурируемые сетевые ресурсы.

ПОДХОДЫ К РЕАЛИЗАЦИИ IaaS

- ⊙ Частные организации, планируя облачную инфраструктуру, чаще всего приходят к одной из следующих двух моделей облачных систем:
 - ⊙ **Виртуализации центров обработки данных:** с одной стороны, есть компании, которые понимают облако как расширение виртуализации в центрах обработки данных, поэтому ищут решение на подобии VMware vCloud Suite - как инструмент автоматизации инфраструктуры, чтобы организовать и упростить управление виртуализированными ресурсами.
 - ⊙ **Предоставление инфраструктуры:** с другой стороны, есть компании, которые понимают облако как “Частный Amazon Web Services (AWS)”, поэтому ищут инструмент резервирования и предоставления виртуализированных ресурсов по требованию.

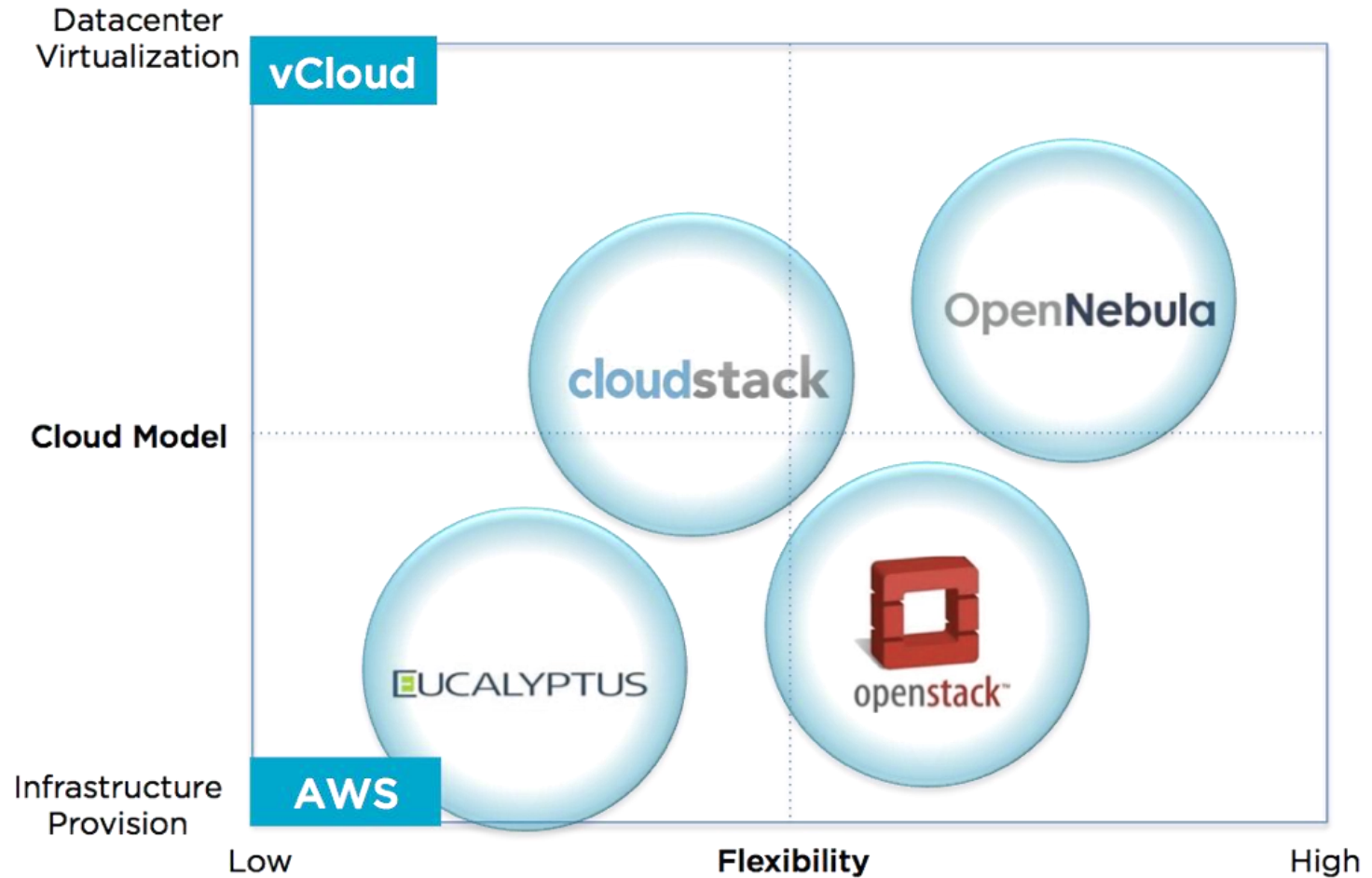
ПОДХОДЫ К РЕАЛИЗАЦИИ IaaS

Виртуализация ЦОД: создаем инфраструктуру для поддержки приложений

Предоставление инфраструктуры: создаем приложения, соответствующие инфраструктуре

	Виртуализация ЦОД	Предоставление инфраструктуры
Приложения	Многоуровневые приложения, созданные в традиционном, "корпоративном" стиле	"Перепроектированные" приложения, соответствующие облачной парадигме
Интерфейсы	Сложный API со множеством функций, административный портал	Простой облачный API, портал самообслуживания
Возможность управления	Полное управление жизненным циклом виртуальных и физических ресурсов	Упрощенный процесс управления жизненными циклами ресурсов, абстрагирующий базовую инфраструктуру
Развертывание облака	Частное	Публичное
Внутренний дизайн	Дизайн «снизу вверх», для управления сложностью ЦОД	Дизайн «сверху вниз» для эффективной реализации облачных интерфейсов
Корпоративные возможности	Высокая доступность, отказоустойчивость, репликация, планирование ... обеспечивается платформой управления облаком	Большинство из них встроено в приложение, спроектированное по принципу "дизайн для максимальной надежности"
Интеграция с ЦОД	Простота адаптации, обеспечивает интеграцию в любую существующую инфраструктуру	Создается на новой, однородной инфраструктуре

КЛАССИФИКАЦИЯ ОБЛАЧНЫХ РЕШЕНИЙ



The VMware vCloud logo is centered in a white square with a subtle gradient. It consists of the word "vmware" in a lowercase, sans-serif font, followed by a registered trademark symbol (®). Below it, the word "vCLOUD" is written in a bold, uppercase, sans-serif font.

vmware®
vCLOUD

VMWARE vCLOUD

VMWARE vCLOUD

8

VMware vCloud Suite

Управление

VMware vCenter
Operations
Management Suite

VMware vFabric
Application
Director

Облачная инфраструктура

VMware vCloud Director

Программная сеть
и система безопасности

VMware vCloud
Networking and Security

Программное хранилище
и обеспечение доступности

VMware vCenter
Site Recovery Manager

Расширяемость

API-интерфейсы
VMware vCloud

VMware vCloud
Connector

VMware vCenter
Orchestrator

Виртуализация

VMware vSphere

Физическая инфраструктура
(сервер, хранилище, сеть)

VMWARE vCLOUD

- ◎ Базирuется на гипервизоре vSphere
- ◎ **vCloud Director** обеспечивает виртуализацию ЦОД
- ◎ **vCloud Connector** обеспечивает интеграцию с общедоступными облаками
- ◎ **vCloud Networking** - программные сетевые компоненты и система безопасности
- ◎ **vCenter Site Recovery Manager Enterprise** -автоматизированное планирование, тестирование и выполнение аварийного восстановления
- ◎ **vCenter Operations** - мониторинг производительности и оповещения; планирование и оптимизация ресурсов; создание отчетов
- ◎ **VMware vCenter Chargeback Manager™** - система внутренних расчетов за потребленные ресурсы



APACHE CLOUDSTACK

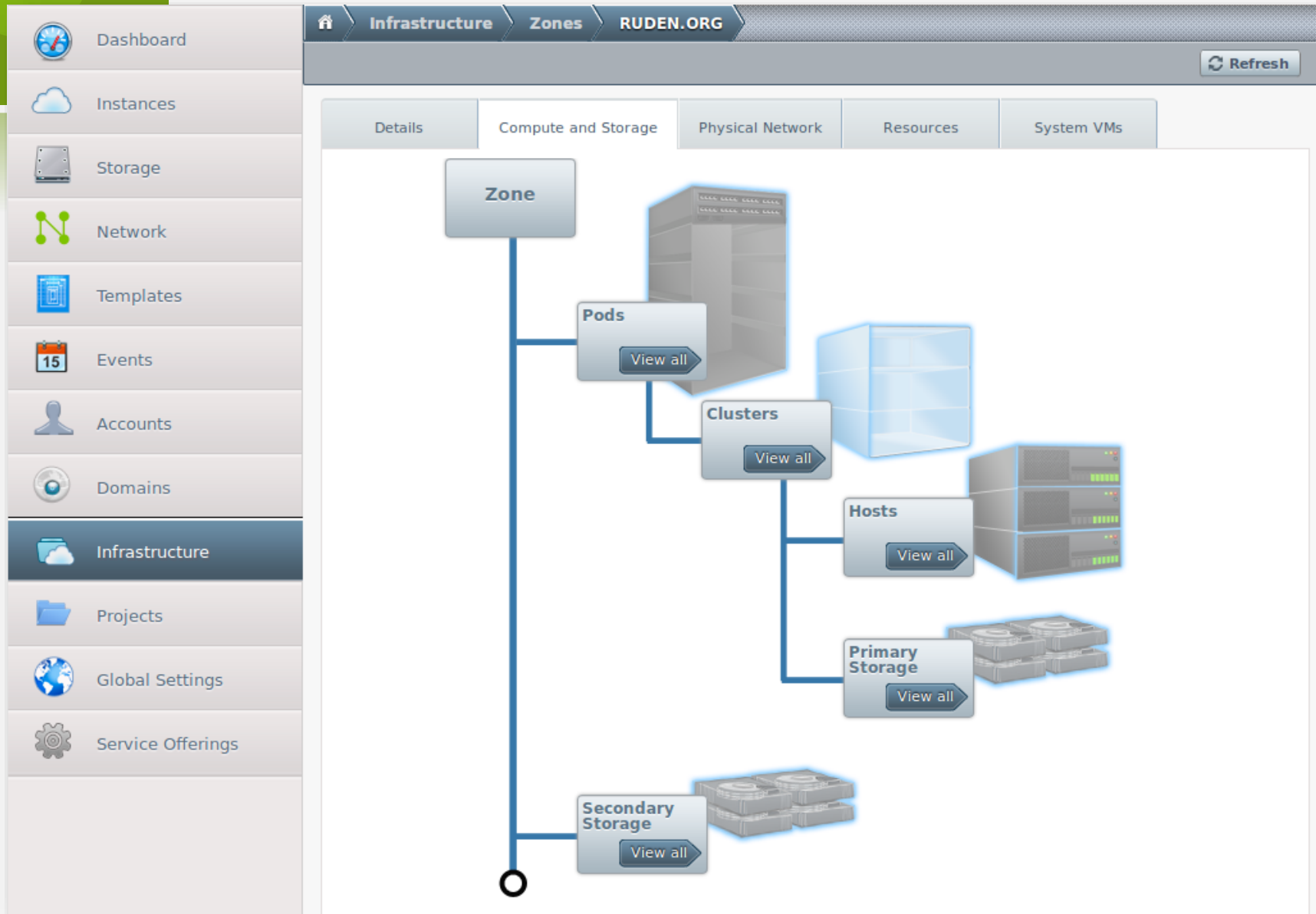
Александр Руденко. CLOUDSTACK 4. АРХИТЕКТУРА, ОСОБЕННОСТИ И НЕДОСТАТКИ.
Системный Администратор, Апрель 2013 г. [<http://ivirt-it.ru/2013/04/cloudstack-4-final/>]

CLOUDSTACK

- ◎ Cloudstack — это консоль управления вычислительными ресурсами датацентра.
- ◎ На этой платформе построили свои облака такие крупные компании как Zynga, Nokia Research Center, Cloudcentral и другие.
- ◎ Развитие проекта осуществлялось компанией Citrix. В 2012 году проект был передан Apache Software Foundation для развития.
- ◎ У платформы есть свой API, который обеспечивает настройку и интеграцию платформы с имеющейся инфраструктурой, а с помощью переходника **CloudBridge Amazon EC2** можно конвертировать Amazon API в Cloudstack API.
- ◎ Обеспечивает управление пользователями, учет ресурсов облачной среды, управление ресурсами ЦОД, сетевыми ресурсами и др.

УРОВНИ ОБЛАКА CLOUDSTACK

12



УРОВНИ ОБЛАКА CLOUDSTACK

- ◎ **Зона (Zone)** — является крупнейшей структурой в иерархии CloudStack. Зона обычно соответствует одному центру обработки данных, хотя допустимо иметь несколько зон в рамках одного центра обработки данных.
- ◎ **Стойка (Pod)** - является второй по величине структурой в иерархии CloudStack и представляет из себя аналог физической стойки с серверами. Стойки, содержатся в зонах. Каждая зона может включать в себя одну или несколько стоек.
- ◎ **Кластер** — это группа физических серверов(хостов) с однотипной конфигурацией размещенных в одной стойке. Все хосты должны работать под управлением однотипного гипервизора, находится в одной подсети и иметь доступ как минимум к одному общему хранилищу.
- ◎ **Узел (Host)** — физический сервер с установленным гипервизором (KVM, Xen или ESXi). Узлы(далее хосты) обеспечивают вычислительные ресурсы, на которых работают ВМ. Хосты являются самой маленькой единицей в инфраструктуре CloudStack.

УПРАВЛЕНИЕ РЕСУРСАМИ

- ⊙ Выделение практически любых ресурсов, будь то процессор, память, дисковое пространство или сетевые ограничения производится путем применения шаблонов (Offerings) предварительно определенных администратором. Существует несколько категорий шаблонов:
 - ⊙ **Compute Offerings** — описывает основные ресурсы VM, такие как максимальная чистота и количество CPU, объем памяти, место хранения VM (локально или на общем хранилище).
 - ⊙ **System Offerings** — описывает наборы ресурсов для системных VM (System Vms).
 - ⊙ **Disc Offerings** — предустановки для дисков VM. Размеры и описание.
 - ⊙ **Network Offerings** — Здесь можно установить ограничение на пропускную способность, VLAN, разрешить или запретить определенный тип трафика.

С ТОЧКИ ЗРЕНИЯ ПОЛЬЗОВАТЕЛЯ

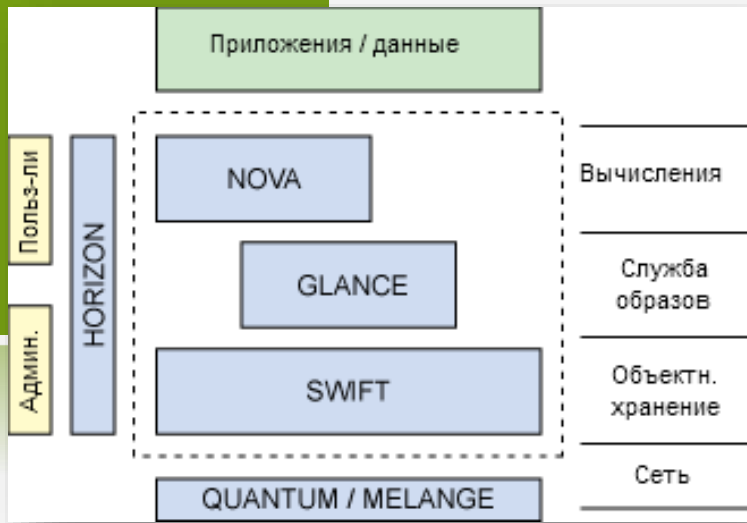
- ☉ Для выделения виртуальной инфраструктуры группе пользователей в CloudStack реализованы проекты (Projects).

The screenshot displays the CloudStack Project dashboard. The interface includes a top navigation bar with 'Notifications', 'Default View', 'Support', and the user name 'Алекс Руд'. A left sidebar contains navigation links for Dashboard, Instances, Storage, Network, Templates, and Events. The main content area is titled 'Project dashboard' and features a 'Dashboard' tab. The 'Compute and Storage' section shows three metrics: Virtual machines (0 Running, 0 Stopped), Storage (0 Volumes), and Bandwidth (200 mb/s). The 'Networking and security' section lists 0 IP Addresses, 0 Load balancing policies, and 0 Port forwarding policies. The 'Users' section shows two users: Support2 and support1. The 'Events' section displays a list of recent activities, including user logins and logouts, and a scheduled async job for updating a project.



OPENSTACK

М. Тим Джонс, Облачные вычисления и хранение данных с помощью OpenStack.
[<http://www.ibm.com/developerworks/ru/library/cl-openstack-cloud/>]



- ◎ **OpenStack** — относительно новый проект IaaS, его первая версия вышла в конце 2010 года.
- ◎ **OpenStack** — набор пакетов ПО с открытым исходным кодом (включая ядро и разрастающиеся проекты), которые в совокупности образуют мощный и зрелый стек IaaS:
 - ◎ **Horizon**: пользовательский интерфейс для управления услугами OpenStack как для пользователей, так и для администраторов
 - ◎ **Nova**: масштабируемая вычислительная платформа, которая обеспечивает подготовку большого числа серверов и виртуальных машин и управление ими (независимо от гипервизора)
 - ◎ **Swift**: масштабируемая объектная система хранения данных
 - ◎ **Quantum** и **Melange**: обеспечивают сетевую связь в виде услуг.
 - ◎ **Glance**: хранилище образов виртуальных дисков (образы как услуги).

УСТАНОВКА И НАСТРОЙКА

- ⦿ Процесс установки и настройки такого комплекса ПО «с нуля» довольно затруднителен.
- ⦿ Поэтому для «попробовать» можно воспользоваться образами DevStack, которые можно развернуть непосредственно на домашнем ПК:

<http://www.openstack.org/software/start/>

ubuntu[®] OpenStack Dashboard

Project: Admin

PROJECT: demo

Manage Compute

Overview

Instances & Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Instances & Volumes

Success: Instance "demo1" launched.

Launch Instance Terminate Instance

Instance Name	IP Address	Size	Status	Task	Power State	Actions
demo1	10.1.0.3 172.16.1.1	512MB RAM 1 VCPU 0 Disk	Active	None	Running	Edit Instance

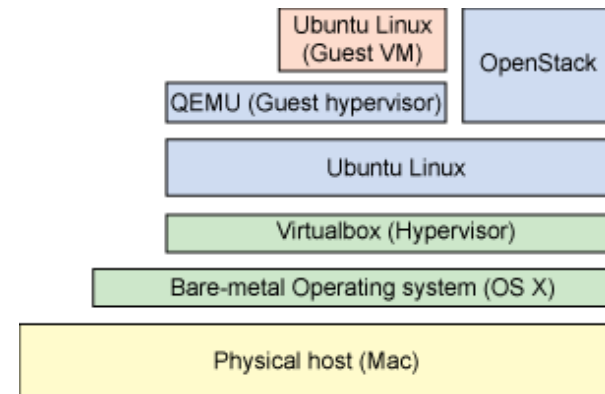
Displaying 1 item

Volumes

Create Volume

Name	Description	Size	Status	Attachments	Actions
No items to display.					

Displaying 0 items





0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0

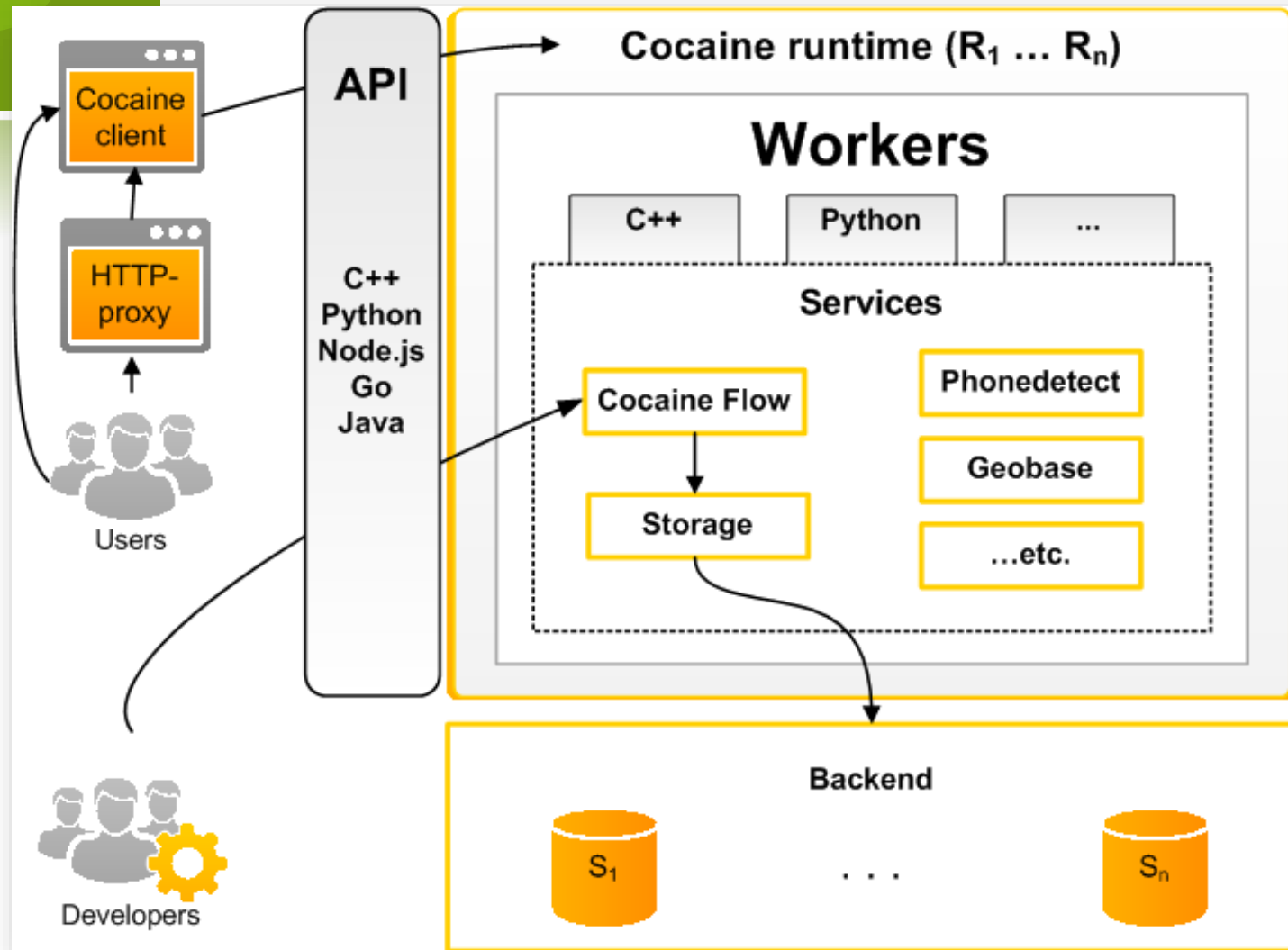
ЯНДЕКС КОКАИН

<http://api.yandex.ru/cocaine/>

- ◎ Yandex Cocaine (**C**onfigurable **O**mnipotent **C**ustom **A**pplications **I**ntegrated **N**etwork **E**ngine) — это PaaS-система (Platform-as-a-Service) с открытым исходным кодом, позволяющая создавать собственные облачные хостинги приложений, такие как Google AppEngine или Heroku.
- ◎ Любую библиотеку или службу можно реализовать в Cocaine как сервис, для этого существует специальное API. Уже реализовано несколько наиболее необходимых. Например: сервис определения региона или языка пользователя, сервис для доступа к хранилищу MongoDB или url-fetcher.
- ◎ Облако состоит из одной и более независимых машин, на которых установлен сервер Cocaine (Cocained).

АРХИТЕКТУРА СОСАINE

21



АРХИТЕКТУРА СОСАИНЕ

- ⊙ Пользователи не знают ничего о местонахождении сервисов, к которым они обращаются, — известен только адрес балансировщика и имя приложения.
- ⊙ Пользовательский запрос отправляется в балансировщик, который передаёт его в облако. В облаке выбирается оптимальная для запроса машина, на которой он и исполняется.
- ⊙ Разработчику необходимо отправить свой код в облако и описать в специальном манифесте, как этот код исполнять.
- ⊙ В качестве технологии для виртуализации инфраструктуры и изоляции приложений используется проект **Docker** — это открытая технология, предоставляющая способ для создания легковесных, переносимых и самодостаточных контейнеров из любых приложений.
- ⊙ В отличие от сред с полной виртуализацией, таких как Xen или KVM, контейнеры используют общее ядро и не предоставляют возможности эмуляции устройств, но при этом их использование не влечёт дополнительных накладных расходов и запускаются они практически мгновенно.



ПРИМЕНЕНИЕ COSAINE

- ⊙ В облаке можно запускать приложения, написанные на различных языках программирования (C, C++, Python, JavaScript и др.). Cosaine в связке с веб-сервером позволяет получить автоматически масштабируемую среду исполнения веб-приложений.
- ⊙ Облако удобно использовать для обработки больших объемов информации — благодаря потоковому интерфейсу обмена данными.
- ⊙ С помощью Cosaine легко реализовать отказоустойчивую и высокопроизводительную шину передачи сообщений и событий. Использование балансировщика нагрузки позволяет создавать отзывчивые к действиям пользователя системы.
- ⊙ Приложения в облаке исполняются независимо друг от друга. Это позволяет поддерживать несколько версий одного и того же приложения и организовывать «мягкую» миграцию пользователей на обновленные версии продуктов.
- ⊙ В облаке можно тестировать приложения, запуская их в изолированном окружении. Так ошибки в программах не повлияют на физическую систему или другие приложения.