Behind the Clouds

Eric Bezille
Chief Technologist
Sun Microsystems France, SAS

All Clouds Share Key Traits

Cloud Computing Layers

Software as a Service (SaaS)

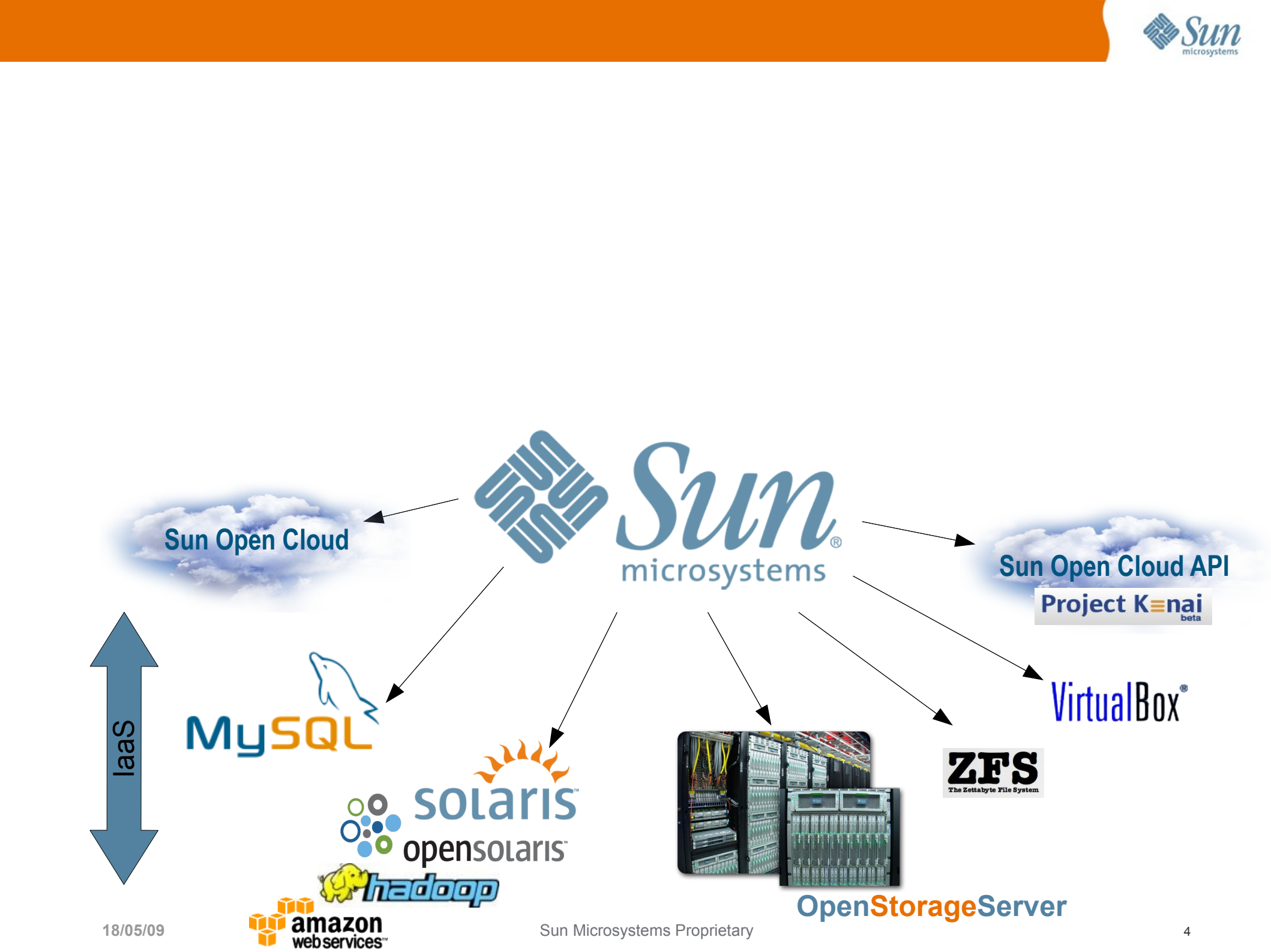
Applications offered on-demand over the network (salesforce.com)

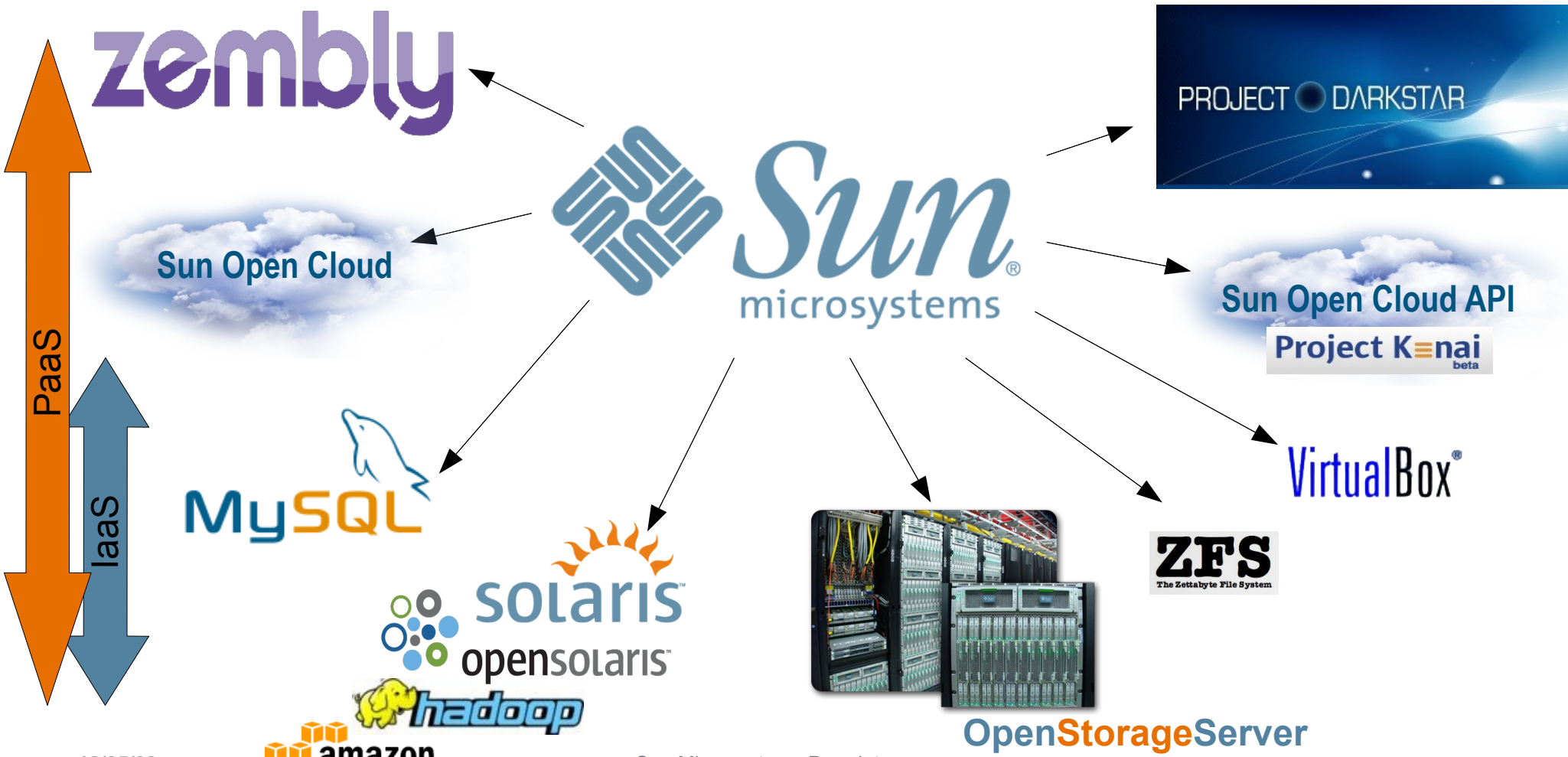
Platform as a Service (PaaS)

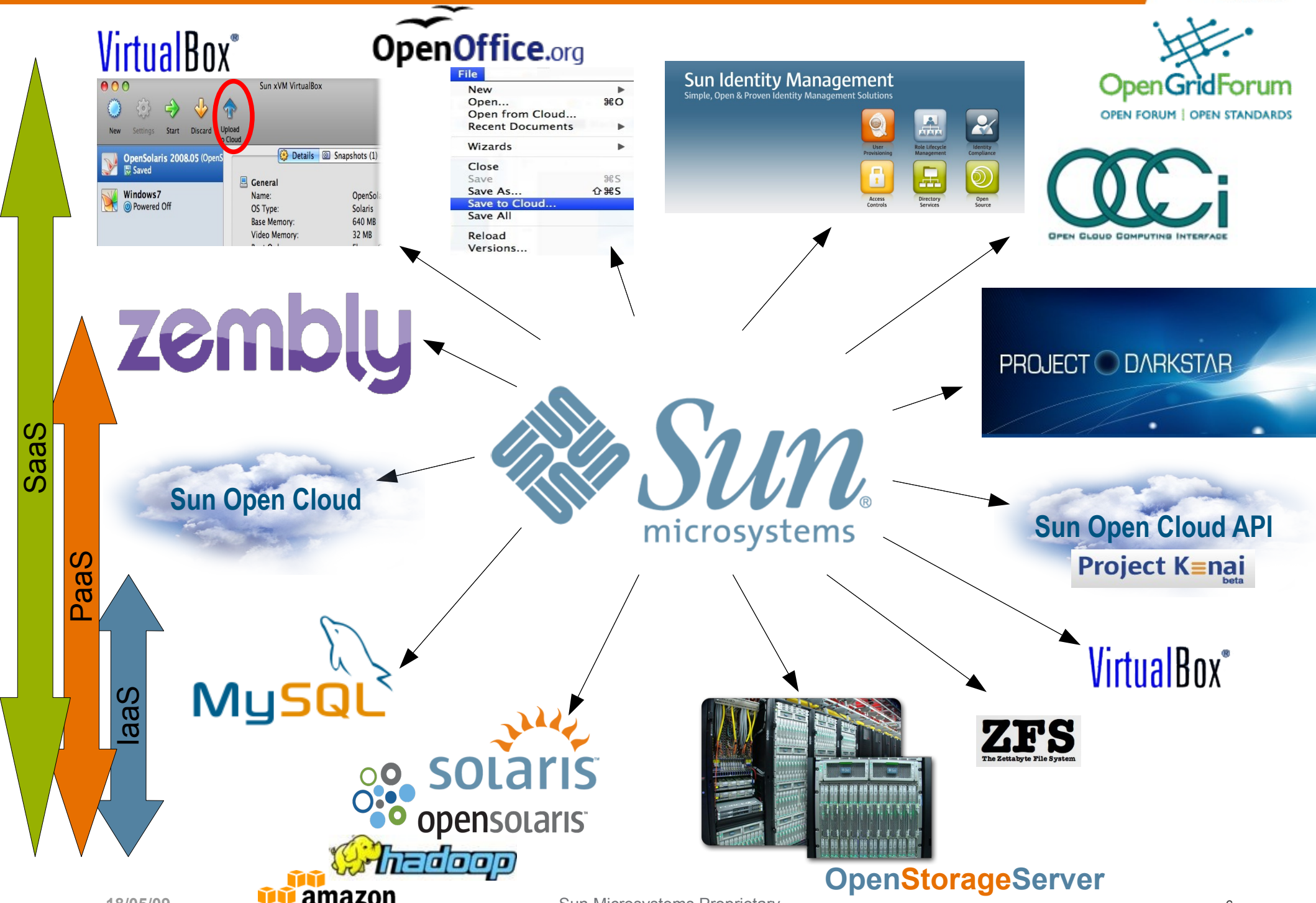
Developer platform with built-in services (Google App Engine)

Infrastructure as a Service (IaaS)

Basic storage and compute capabilities offered as a service (Amazon web services)

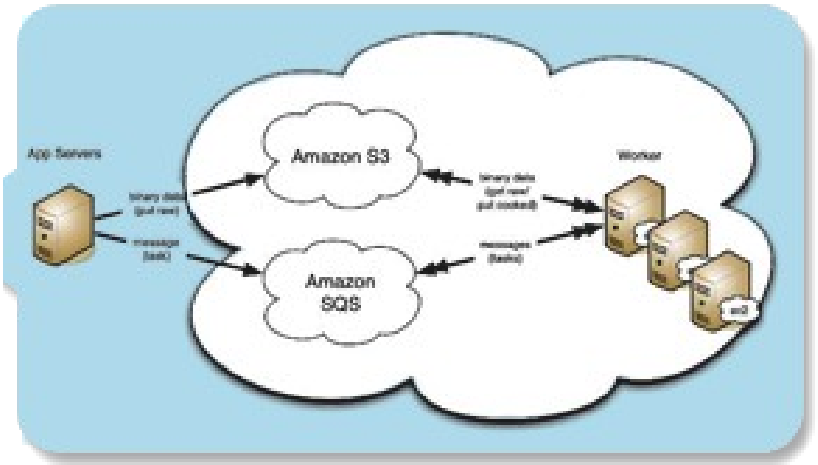

Sun Open Cloud
Sun microsystems
Sun Open Cloud API
Project Kenai
beta
IaaS
MySQL
VirtualBox
solaris
opensolaris
hadoop
amazon web services
ZFS
The Zettabyte File System
OpenStorageServer


zembly
PROJECT DARKSTAR
Sun Open Cloud
Sun microsystems
Sun Open Cloud API
Project Kenai
beta
PaaS
IaaS
MySQL
VirtualBox
solaris
opensolaris
ZFS
The Zettabyte File System
hadoop
amazon web services
OpenStorageServer


VirtualBox
OpenOffice.org
Sun Identity Management
Simple, Open & Proven Identity Management Solutions
OpenGridForum
OPEN FORUM | OPEN STANDARDS
OCCI
Open Cloud Computing Interface
zembly
PROJECT DARKSTAR
SaaS
PaaS
IaaS
Sun Open Cloud
Sun microsystems
Sun Open Cloud API
Project Kenai
beta
MySQL
VirtualBox
solaris
opensolaris
hadoop
amazon web services
ZFS
The Zettabyte File System
OpenStorageServer

Example : SmugMug.com

- Professional Photo Site
- Sun Systems for Web, MySQL & 1st Tier Storage
- Amazon S3 & EC2 for archive
- Functional offload to cloud
- Many modern Web 2.0 startups and SMB businesses use similar model



Don MacAskill

SmugMug's CEO & Chief Geek

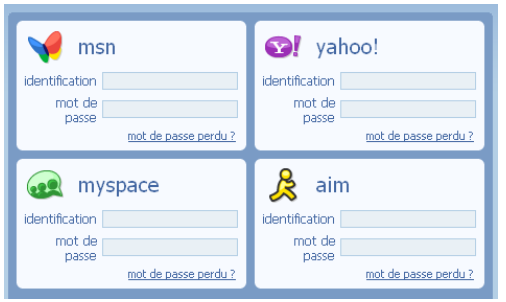
http://blogs.smugmug.com/don/2008/10/10
http://blogs.smugmug.com/don/2008/12

Cloud Social Application Platform

End-Users

facebook

Google



zembly

Your Network

Your API

Online IDE

Developers

Sun Cloud

http://zembly.com/

Login

PROJECT DARKSTAR

Home | Downloads | Library | Core Technology | Projects/Apps | Discussion | Products/Services | News/Events

Open Source for the Online Game Universe

Project Darkstar aims to help developers and operators avoid a range of serious, yet typical, problems associated with **online games, virtual worlds, and social networking applications today**, including zone overloading, data corruption, and server under-utilization. It will also enable developers to support new dimensions of play such as evolving virtual worlds and very large scale battlefields.

DOWNLOAD

Get the latest distribution of Project Darkstar

Curious about Project Darkstar?

LEARN

Ready to try out or use the technology?

START

Want to get involved with the Community?

PARTICIPATE

News & Featured Content

Project Darkstar v0.9.9 Now Available

(April 7, 2009) The Project Darkstar development team is pleased to announce the availability of the Project Darkstar v0.9.9 distribution release. This release introduces support for pluggable client session protocol and transport layers, giving developers new communications options: now, developers can choose to use Project Darkstar's default...[read more]

View All News & Events >

Latest Forum Posts

- Re: Question about updates and results of that by devedse
  May 12, 09, 09:55 pm
- Re: Question about updates and results of that by Jeff
  May 12, 09, 04:25 pm
- Re: Question about updates and results of that by devedse
  May 12, 09, 03:39 pm
- Re: Project Darkstar Developers - Possibly show your games at JavaOne! by

View All Forum Posts >

Quick Links

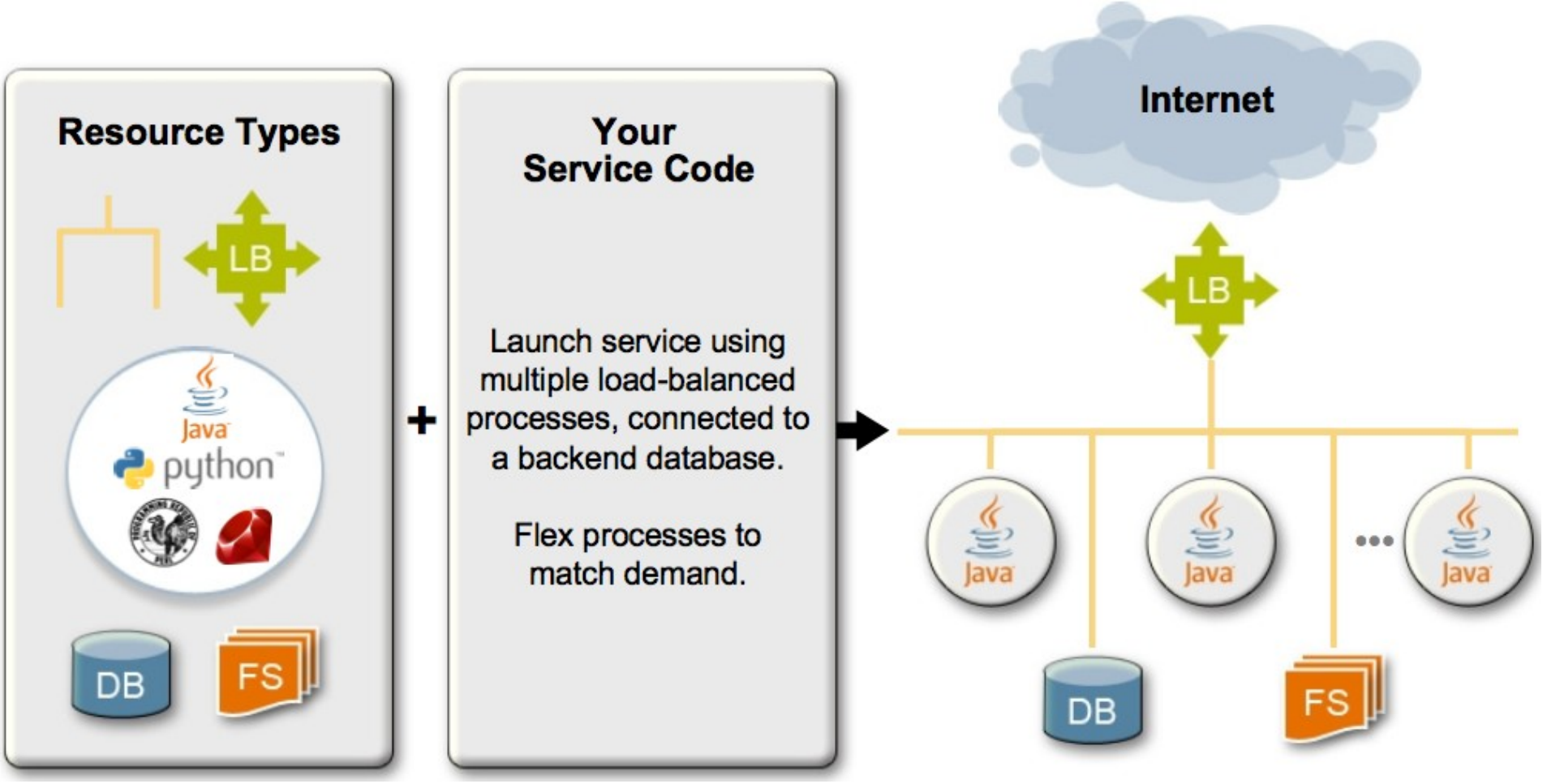
- Community Forum
- FAQ
- Technology Roadmap
- Events Calendar
- Showcase
- Darkstar-based Applications

Sponsored by Sun Labs

Project Caroline

https://www.projectcaroline.net/

Project Caroline helps software providers develop services rapidly, update in-production services frequently, and automatically flex their use of platform resources to match changing runtime demands.



Sun Project – Immutable Service Containers

Why Immutable Service Containers (ISC) on clouds?

For Users: secure my applications from tampering
For SysOp: secure my users from contamination
Make it faster and easier to Deploy

- Isolation: One Container == One Service
  - > Only expose the service in the container (no other network ports)
  - > Only run with unique credentials and required privileges
  - > 90% immutable (kernel, directories, files, etc.) within the container
- Powerful and Flexible
  - > credentials, privileges, and even specific filesystem immutability can be customized based on application requirements
- Configure Once, Deploy Everywhere
  - > created from “golden” templates and customized using service classes
  - > “identical” service instances can be created in < 10 seconds
  - > containers for: apache, pen, memcached, ntp, bind, syslog, mysql

http://wikis.sun.com/display/ISC

Sun's Strategy

Develop the core technologies for Sun's Open Cloud Platform

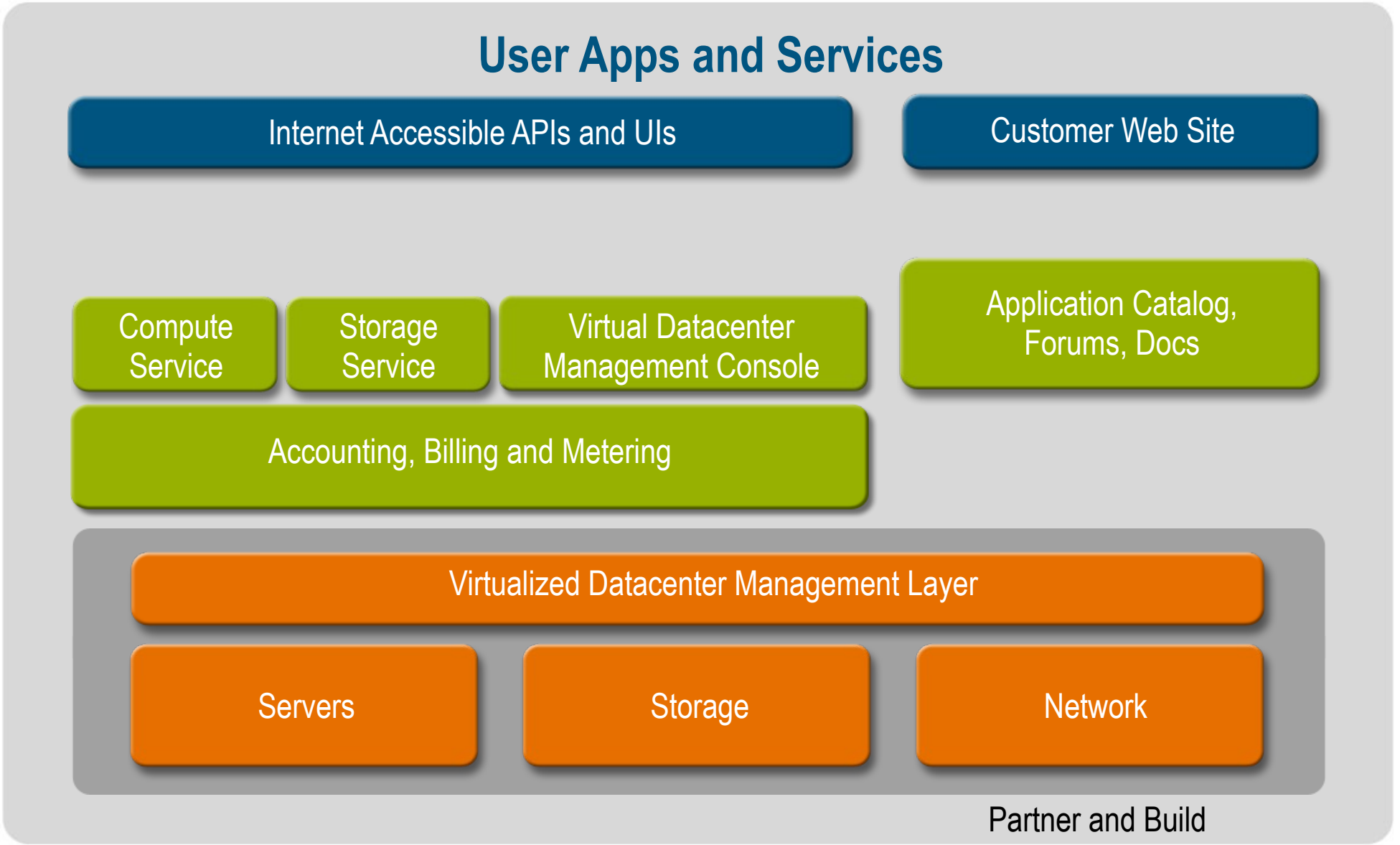
Offer Services through Sun's public cloud service – the Sun Cloud

Work with service providers and enterprises to build their own clouds

Develop open standards

Build partnerships and communities

Cloud Architecture – Phase 1



Storage Service

What It is

- On-demand, API-based access to storage on the network

Features

- Ability to store and retrieve data as objects or files
- REST API with open, AWS S3-like semantics for object storage
- WebDAV for file storage
- Fast and inexpensive cloning of objects and files
- High availability
- Detailed metering of storage used, I/O requests, bandwidth, etc.

Customer Benefit

- Scalable, highly available storage without big hardware investments

Compute Service

What It Is

- On-demand, scalable computing infrastructure accessed via APIs or unique Virtual Datacenter (VDC) UI model

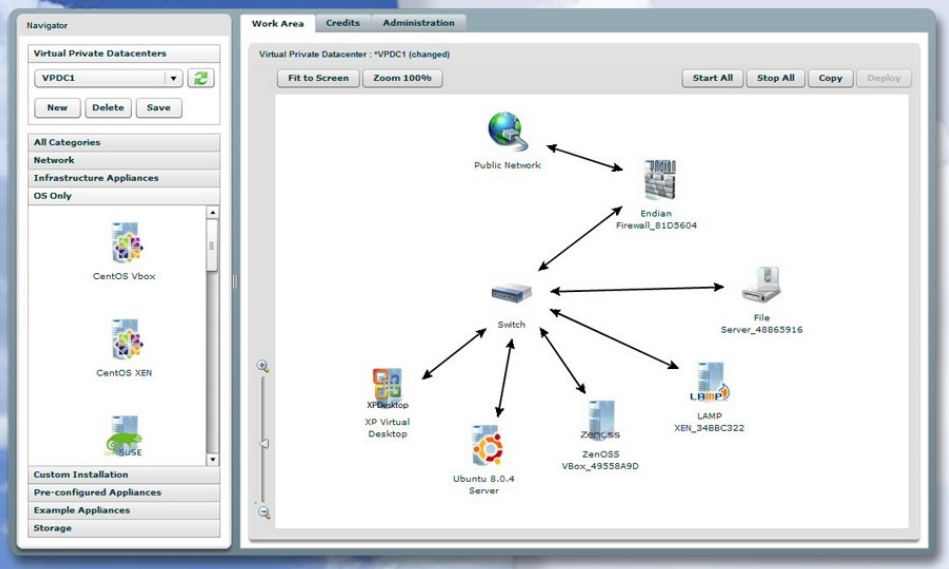
Features

- On-demand provisioning of virtual machines of industry-standard operating systems including Linux, Windows and OpenSolaris
- Control and management with open, AWS EC2-like API or Virtual Datacenter UI
- Creation of custom VMIs and access to pre-configured VMIs in the cloud
- Support for persistent and non-persistent virtual machines

Customer Benefit

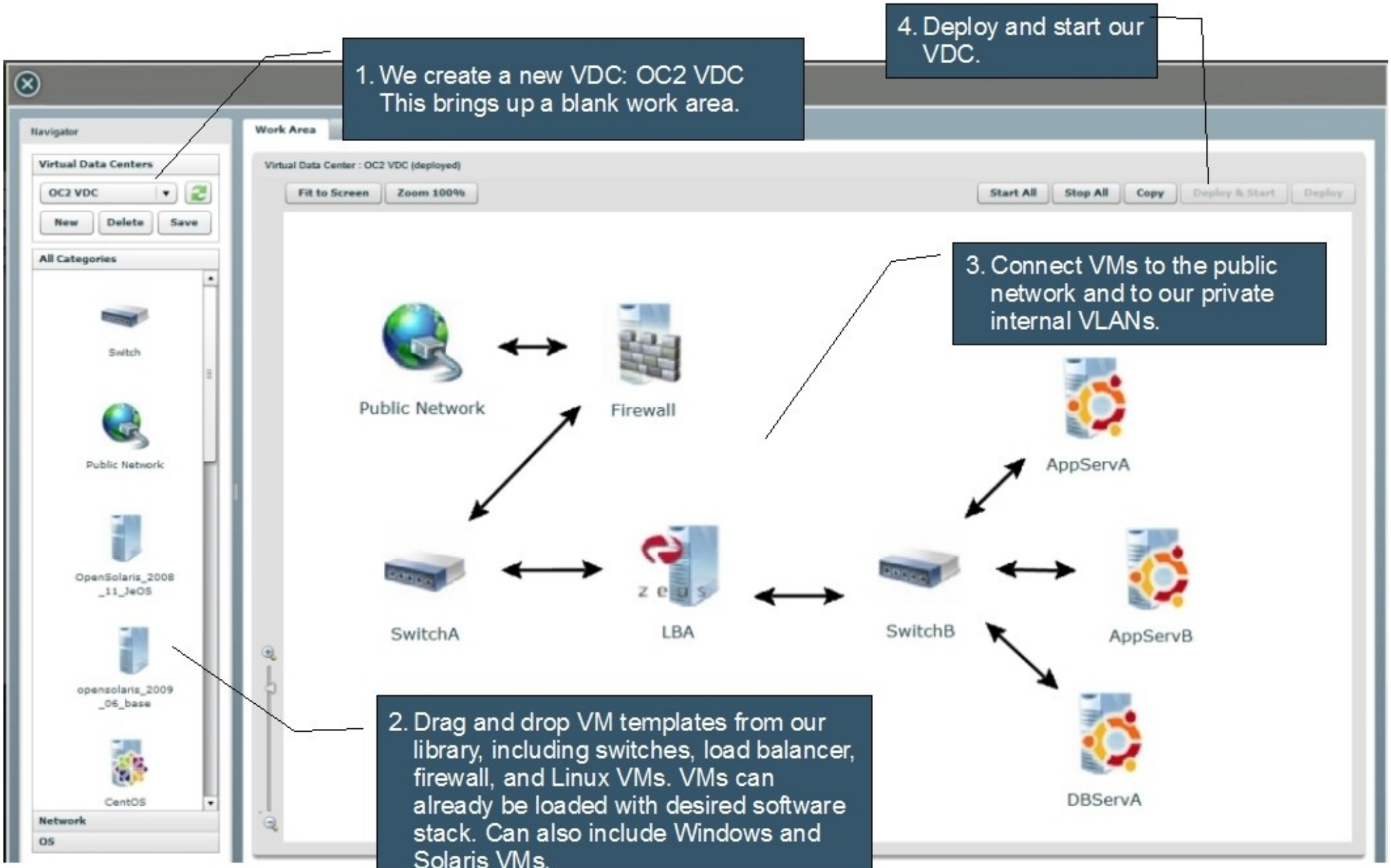
- Affordable access to highly scalable computing infrastructure
- Always available

Sun Virtual Datacenter Model

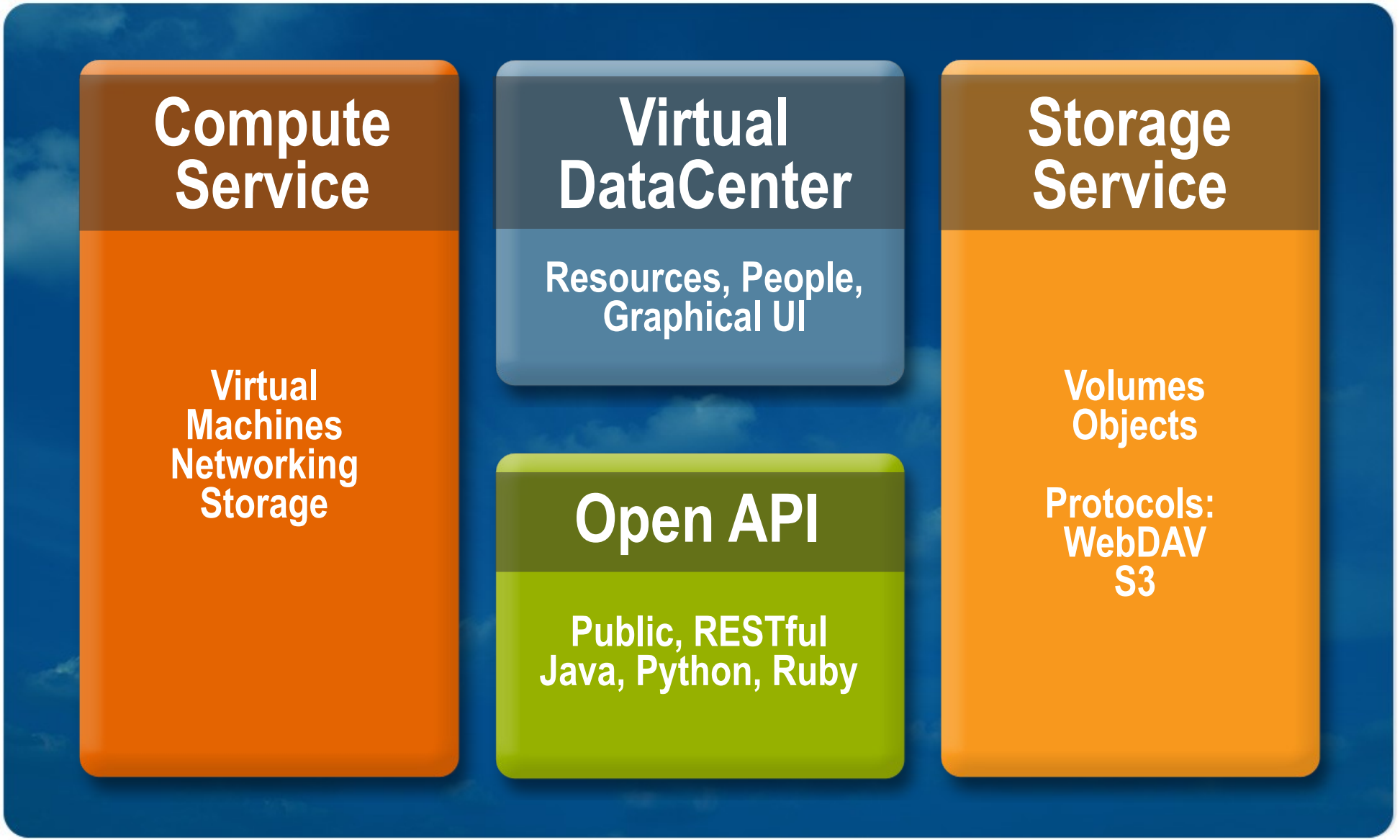


- Design application from pre-built components using drag-and-drop
- Deploy to cloud
- Monitor, manage and reconfigure
- Compatibility with programmatic APIs
- Encapsulate system architecture of an application
- Ability to model, save and deploy entire system

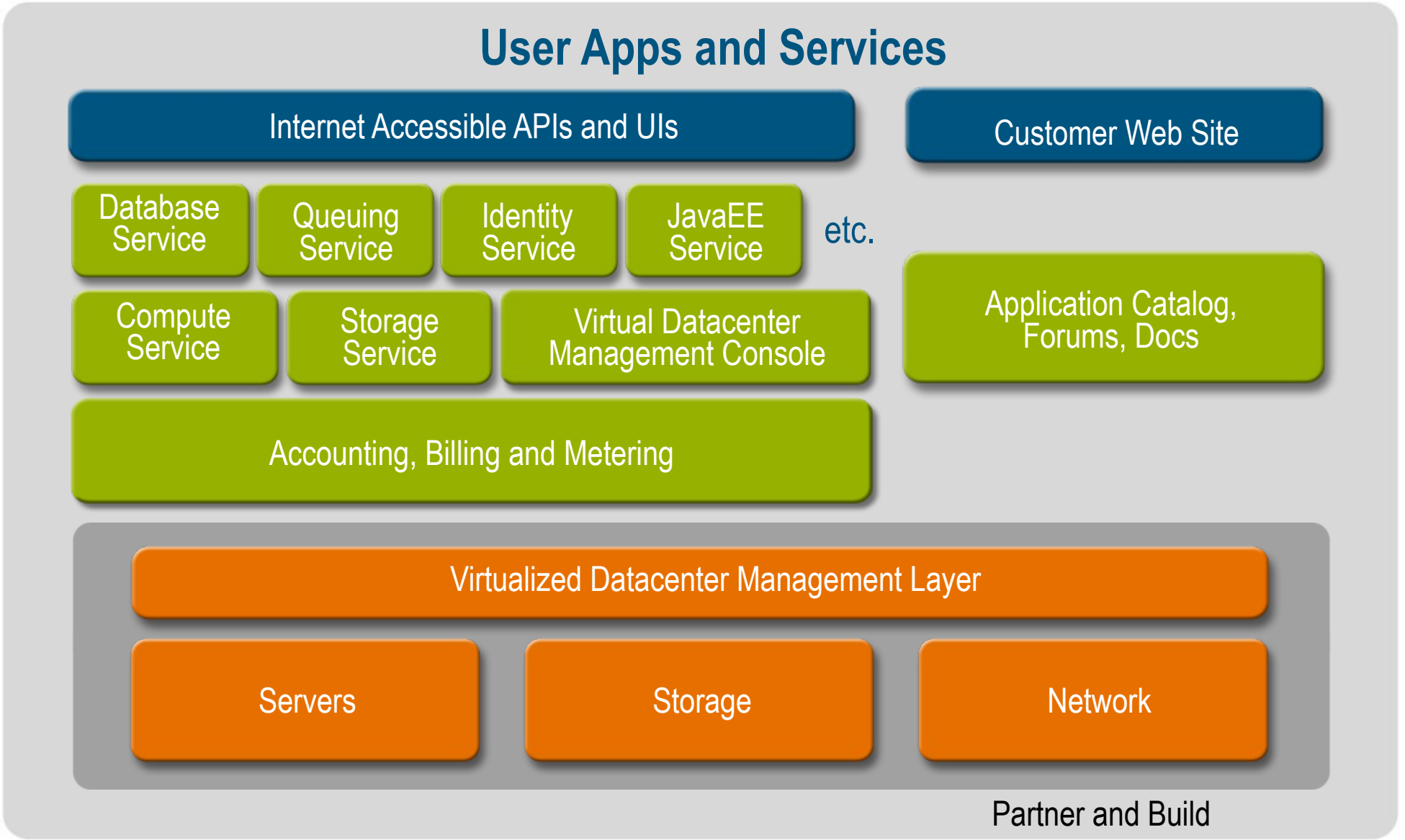
Deployment Steps Example



Sun Cloud Architecture

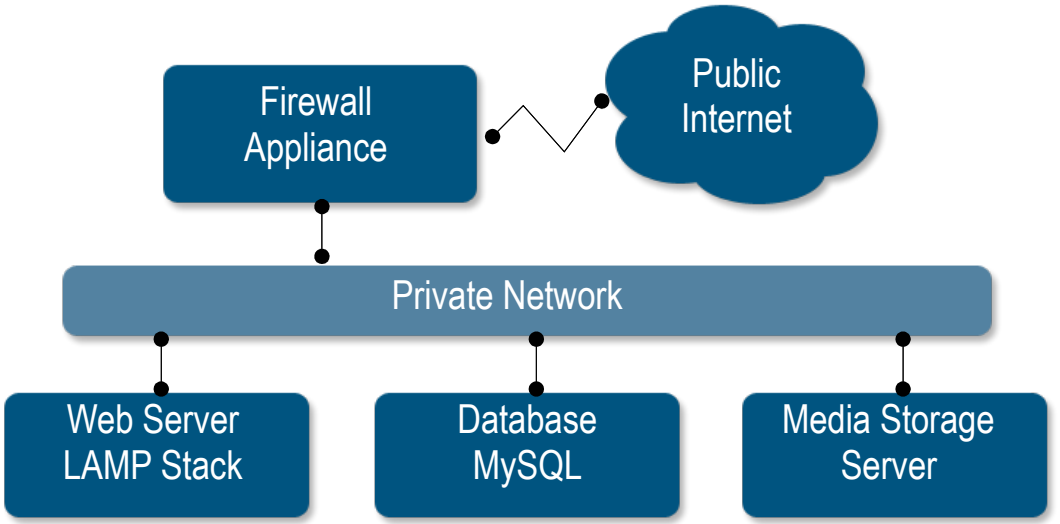


Cloud Architecture – Future



Sun Cloud RESTful API

- Everything is a resource – http GET, POST, PUT...
- Requires only a single starting point - other URIs are discoverable
- Easy to create, save, load, stop, start entire applications
- Released today to the public under Creative Commons



Sun Open Cloud API

Project Kenai beta

Starts with the User's Virtual Data Center

```
GET /xrgy.cloud.sun.com

Server response:

 {
   "name" : "XRGY Virtual Data Center",
   "uri" : "http://xrgy.cloud.sun.com",

   "addresses" : [ ],
   "vnets" : [ ],
   "volumes" : [ ],

   "create-vnet" : "/ops/create-vnet",
   "create-volume" : "/ops/create-vol",
   "create-cluster" : "/ops/create-cluster",

   "clusters" :  [
    {
       "name": "cluster1",
       "uri": "/clusters/cluster1",
       "create-vm" : "/clusters/cluster1/ops/create-vm",
        .......
      }]
```

Note: URI Links

POST to Create a MySQL Virtual Machine

URI of representation for cluster1

```
POST /clusters/cluster1/ops/create-vm

 {
   "name" : "Database"
   "from-template" : "http://cloud.sun.com/resources/vmtemplates/003",
   "description" : "MySQL host",
   "tags" : [ "sql" ]
 }

Server Response

HTTP/1.1 201 Created
 {
   "name" : "Database"
   "uri": "/clusters/cluster1/vms/001",
   "run-status" : "HALTED",
   "description" : "MySQL host",
   "tags" : [ "sql" ]
   "hostname" : "c1vm001",
   ...
   }
```

Example Code Fragment (python)

```
# Create a new Virtual Machine based on an existing one in # the
   application cluster


app_cluster = find_cluster("application")

vm = app_cluster.create_vm({ "name" : "web03", "from-vm" :
   app_cluster.vms[0].uri })


# Attach our new VM to the correct VNets

vm.attach({ "uri" : find_vnet("Front End").uri })

vm.attach({ "uri" : find_vnet("Back End").uri })


# Deploy and start the new VM

vm.deploy()

vm.start()
```

Storage Service Features

WebDAV and Adminstration API
(Volume/Folder/File based access)

- Ability to store and retrieve data as files in folders and volumes
- Ability to clone and snapshot volumes
- Ability to mount file system
- Open, proven and highly scalable protocol

Storage Object API
(AWS S3 compatibility)

- Ability to store and retrieve data as objects in buckets
- REST API with open, AWS S3-like semantics
- S3 compatiblity makes applications portable
- Write, read, and delete objects of any size

> Detailed metering and billing

> Programmatic, fast, highly available & scalable, and inexpensive

Ways to use the services

Administration API

- Java, Ruby and Python libraries
- Comand line tool cURL via shell script
- Sun Cloud Storage Console
- Web UI control

WebDAV Protocol

- Java client library for ease of use
- WebDAV JackRabbit library for additional control
- Cloud Drives
- Existing WebDAV apps

Object API

- Sun's client Java library
- Amazon shell tools
- S3 Fox (coming soon)
- Jets3t Java library (at java,net)
- S3 client application by changing host name

Using the APIs

Project Kenai beta

WebDAV Protocol

Web-based Distributed Authoring and Versioning Protocol is an extension to the HTTP protocol that can be used to create directories and perform file management tasks.

API Calls:

PROPFIND {collection-uri} - Get properties of a collection (i.e. a folder) and potentially all the members of the collection.

PROPFIND {member-uri} - Get properties of a member (i.e. a file)

PROPPATCH {collection-uri} - Set or remove properties of a collection

PROPPATCH {member-uri} - Set or remove properties of a member

MKCOL {collection-uri} - Create a new collection at the specified location

DELETE {collection-uri} - Delete a collection and nested members, along with any locks on those resources

In addition, the usual **GET and PUT** calls

DELETE {member-uri} - Delete a member, along with any locks on that resource

COPY {collection-uri} - Create a copy of a collection and possibly its members

COPY {member-uri} - Create a new copy of a member

MOVE {collection-uri} - Move a collection to a new location

MOVE {member-uri} - Move a member to a new location

LOCK {resource-uri} - Take out a lock on the specified collection or member

UNLOCK {resource-uri} - Release an existing lock on a collection or member

Using the APIs

Project Kenai beta

Administration API

Used in conjunction with the WebDAV service to create volumes, take snapshots and turn the snashots into clones.

API Calls:

Create Volume. Use the POST */service-prefix/resources/volume-name* request with an empty <resource> element in the message body to create a volume.

Create Snapshot. Use the POST */service-prefix/snapshots/volume-name/snapshot-name* request to create a snapshot of a volume.

Get Volumes. Use to get information about an account's volumes and snapshots, or a specific volume by send a GE*T /service-prefix/resources.../volume-name* request.

Create Clone. Use to create a cloneby sending a POST */service-prefix/resources/clone-volume-name* request with a <resource> element in the message body.

Delete Snapshot. Use the DELETE */service-prefix/snapshots/volume-name/snapshot-name* request to delete a snapshot of the specified volume.

Delete Volume. To delete a volume send a DELETE */service-prefix/resources/volume-name* request.

Using the APIs

Project Kenai beta

Storage Object API

Enables the use of existing S3 applications and the ability to create new applications with S3-like semantics

API Calls:

GET All Buckets: Metadata request that retrieves all the bucket information for an account. It is called a *service operation*. .

PUT Bucket: Creates and names a bucket. The bucket names must be unique across all the accounts,

GET Bucket: Metadata request that retrieves all the object information for a bucket.

GET Bucket Location: Metadata request that retrieves the physical location of the bucket. 5.

DELETE Bucket: Deletes the targeted bucket if it is empty.

PUT Object: Creates a new object from a file that is sent over the HTTP protocol.

GET Object: Retrieves an object from the storage service.

HEAD Object: Retrieves header information about an object.

DELETE Object: Deletes an object (The API response is differs from the S3 REST API. See documentation for details.

Sun Open Cloud Platform

OpenStorageServer

VirtualBox

Sun xVM

Q-layer

MySQL

zmanda
Open Source Backup

RIGHTSCALE

Aserver

KickApps

amazon web services

Products and Technologies

Expertise and Services

Partners

Open Communities

TACC

glassfish.dev.java.net

Java

Project Kenai beta

zembly

OpenOffice.org

ZFS
The Zettabyte File System

Eucalyptus

NetBeans

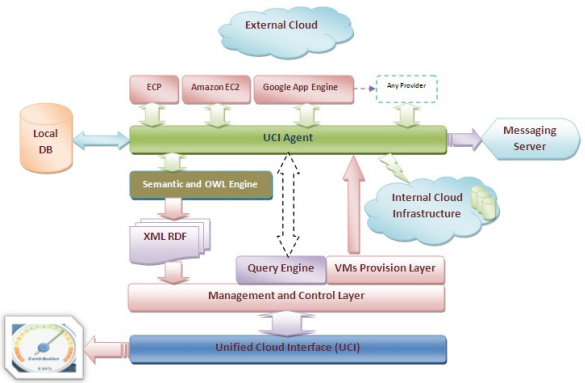
lustre

Roadmap

- First public cloud will rollout starting this in Q2 2009
- Additional services will begin appearing soon after
- Will begin working with customers using product version of software in second half of 2009

Cloud Communities

- Cloud Camps all over the world
- Industry Initiatives
  - > Open Cloud Manifesto
  - > Unified Cloud Interface (UCI)
  - > DMTF Open Cloud Standards Incubator
  - > ...and many more
- Community Projects
  - > Kenai.com Sourceforge for the Cloud:
    - – Sun Cloud APIs, Immutable Service Containers, Dynamic Service Containers, Encrypted Cloud Storage, etc.
  - > Blogs.sun.com, wikis.sun.com



Everyone can participate!

Get Started Today

- Participate in the Development of our Open Cloud APIs – learn more at

  http://kenai.com/projects/suncloudapis/
- Sign up for Early Access to Sun Cloud Services
- Participate in Zembly, DarkStar, ProjectCaroline...
- Become a Sun Cloud Partner
- Let Sun experts help you take advantage of Cloud Computing

http://sun.com/cloud

THANK YOU

eric.bezille@sun.com
http://blogs.sun.com/EricBezille

A Peek Behind the Sun Cloud

VirtualBox®

Sun xVM

TACC

Q-layer

MySQL™

zmanda
Open Source Backup

RIGHT SCALE®

glassfish.dev.java.net

Java™

OpenOffice.org

ZFS
The Zettabyte File System

Eucalyptus

KickApps™

amazon web services™

NetBeans

lustre®

Products and Technologies

Expertise and Services

Partners

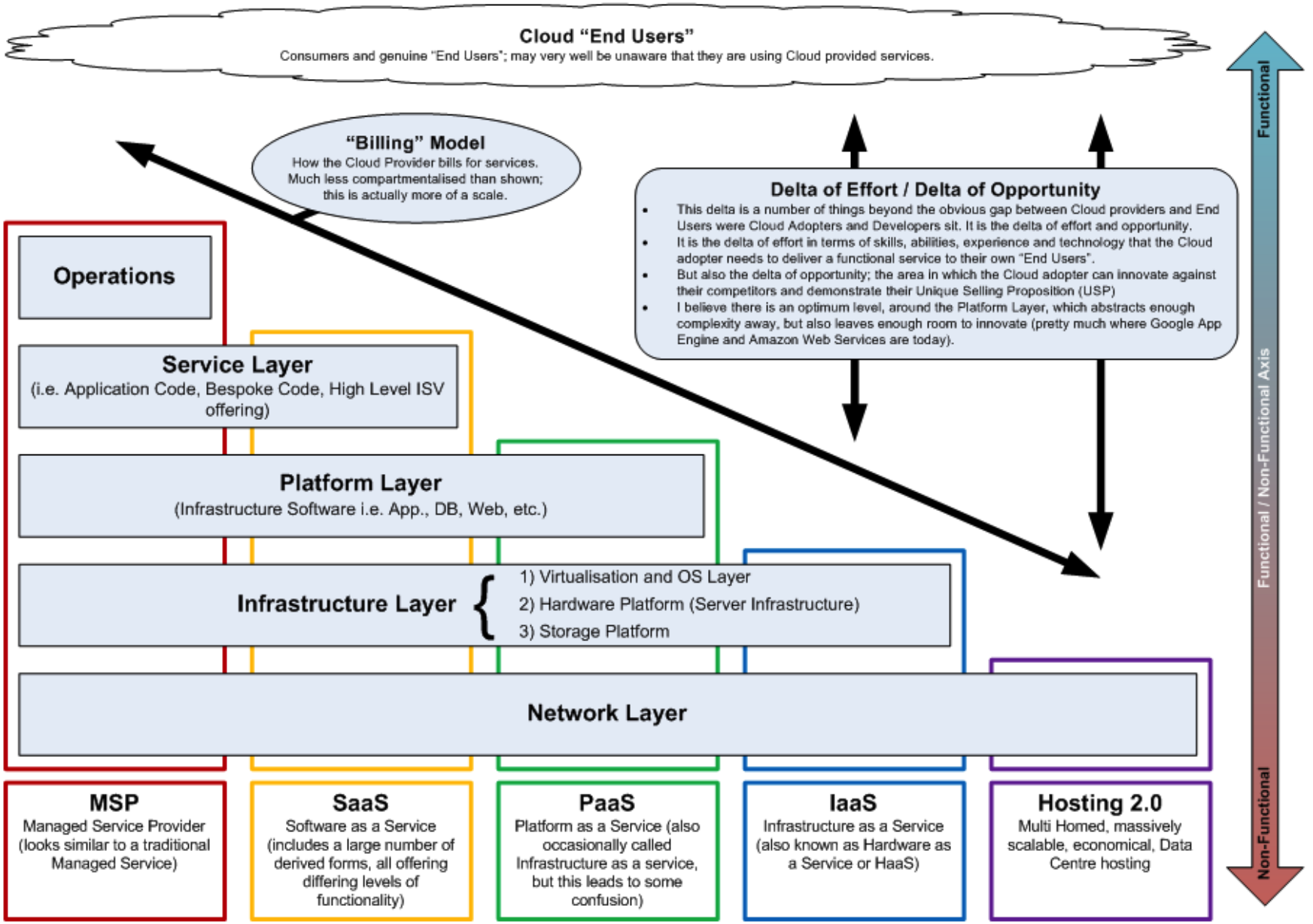
Open Communities

Experts in the room

- Clouds Architectures session : **Constantin Gonzalez**
- Clouds Security session : **Luc Wijns**
- Identity Management in the Clouds : **Didier Burkhalter, Alain Barbier**
- zembly PaaS : **Emmanuel De Lagardette**
- MySQL in the Clouds : **Stéphane Varoqui, Serge Frezefond**
- opensolaris and more... : **Gilles Gravier, Bruno Gillet**
- OpenStorageServer : **Christophe Pauliat, Denis Martinez**
- Startup Essential, to build your Clouds : **Thierry Manfé**

All can be joined at : first.last@Sun.Com

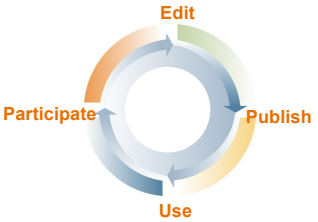
Cloud Computing Layers

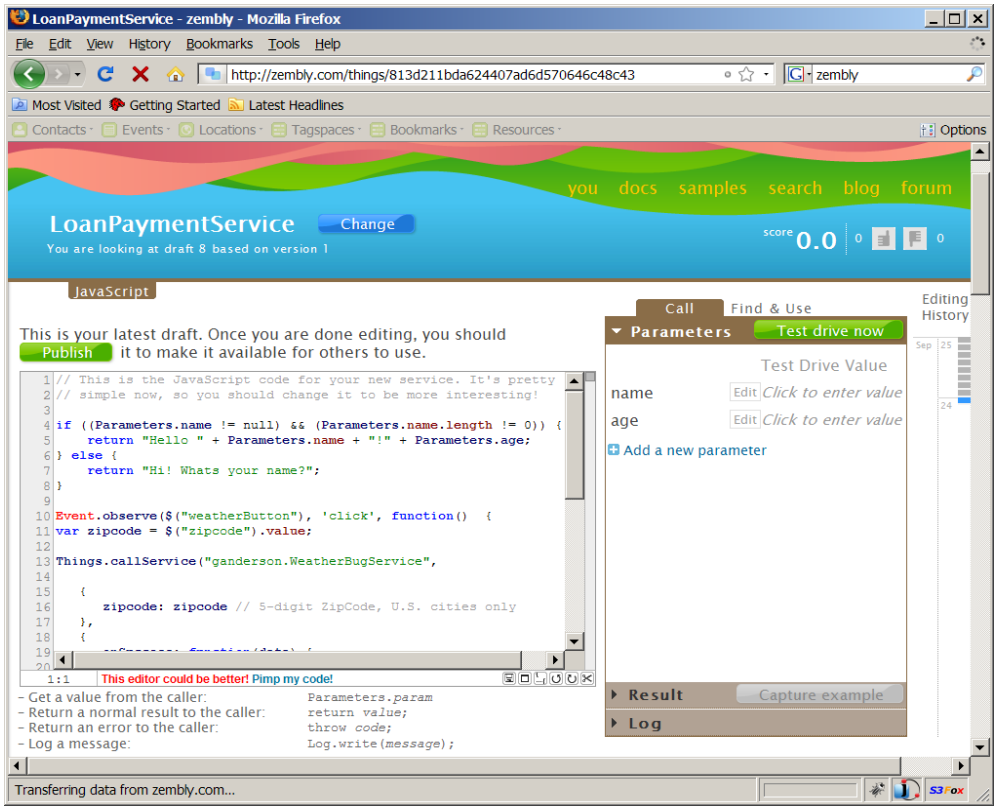


Source : Wayne Horkan blog – Sun Microsystems

Platform as a Service

Develop in the cloud : www.zembly.com





A cloud based development environment

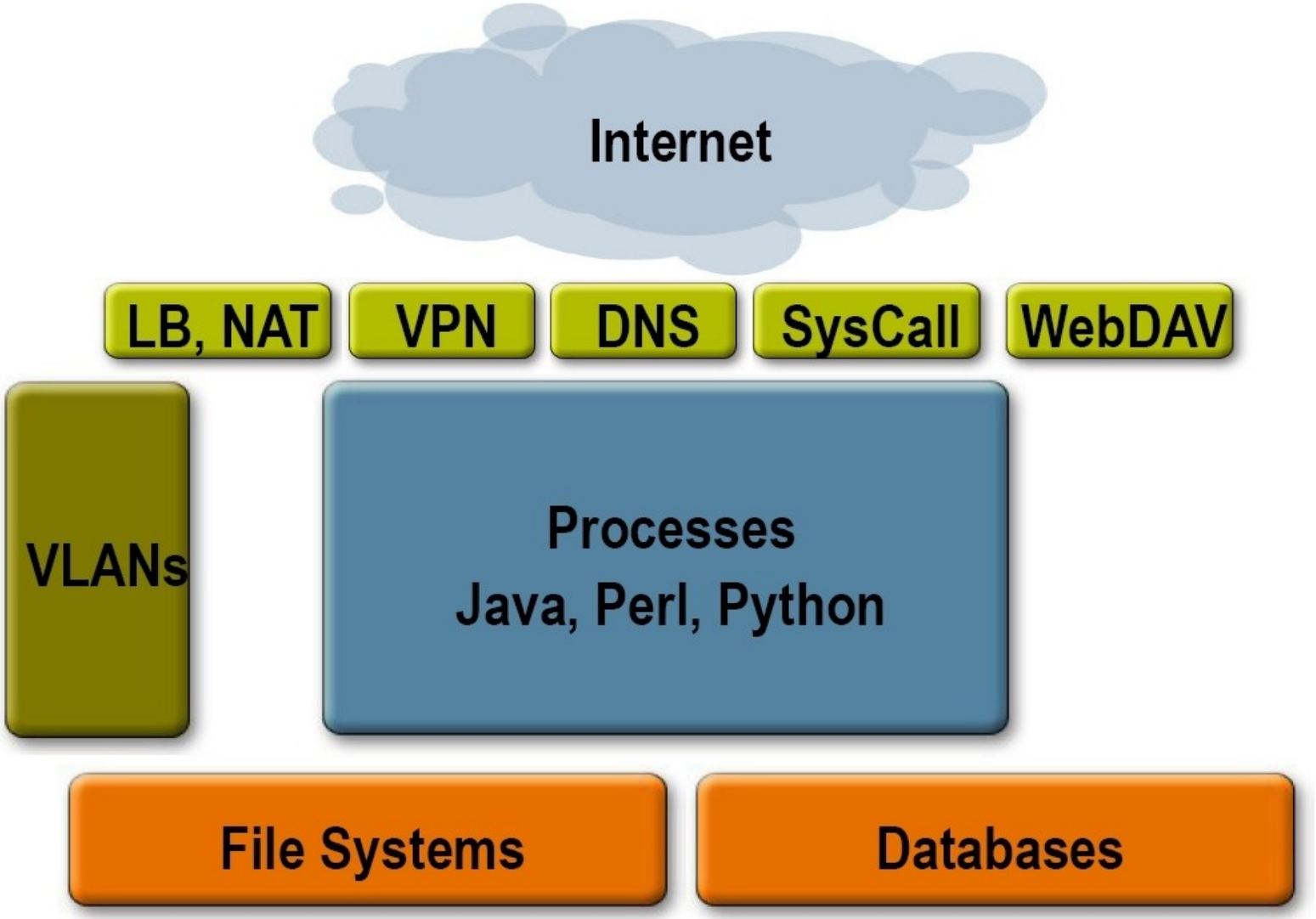
for building the Web for casual developers

Wikipedia for "Live" Code

Social Ecosystem of Contributors and Consumers

Development Platform for Next Generation Web Apps

Project Caroline



Inspect the MySQL Virtual Machine

```
GET /clusters/cluster1/vms/01

Server response:
     {
       "name" : "Database"
       "uri": "/clusters/cluster1/vms/001",
       "description" : "MySQL host",
       "back-up": "/clusters/cluster1/vms/001/ops/back-up",
       "attach": "/clusters/cluster1/vms/001/ops/attach",
       "detach": "/clusters/cluster1/vms/001/ops/detach",

       "controllers": {
         "deploy": "/clusters/cluster1/vms/001/ops/deploy",

         "start": "/clusters/cluster1/vms/001/ops/start",
         "stop": "/clusters/cluster1/vms/001/ops/stop",

         "reboot": "/clusters/cluster1/vms/001/ops/reboot",
         "hibernate":
              "/clusters/cluster1/vms/001/ops/hibernate",
         "resume": "/clusters/cluster1/vms/001/ops/resume"
       }
     }
```

URI of Database Virtual Machine

Deploy the Application

```
POST /clusters/cluster1/ops/deploy

Host: xrgy.cloud.sun.com

{
  "note": "System deployed at 8:00 pm by Jack!"
}
```

URI from representation for cluster1

Start the Application

```
POST /clusters/cluster1/ops/start

Host: xrgy.cloud.sun.com

{
  "note": "System started at 8:05 pm by Jack!"
}
```

Stop the Application

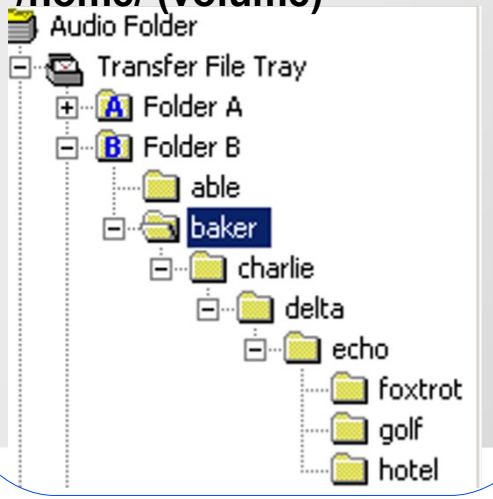
```
POST /clusters/cluster1/ops/stop

Host: xrgy.cloud.sun.com

{
  "note":  "System halted at 10:32pm by Jill!"
}
```

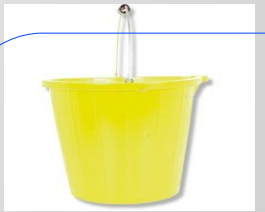
Sun Cloud Storage Service

Vernacular

Volumes and Buckets: Two different ways of creating the top level folder in a file system.
Folder/Collection: Sub-folder in a volume
File/Member: Data in a folder or collection
Objects: Data in a bucket.

/home/ (volume)

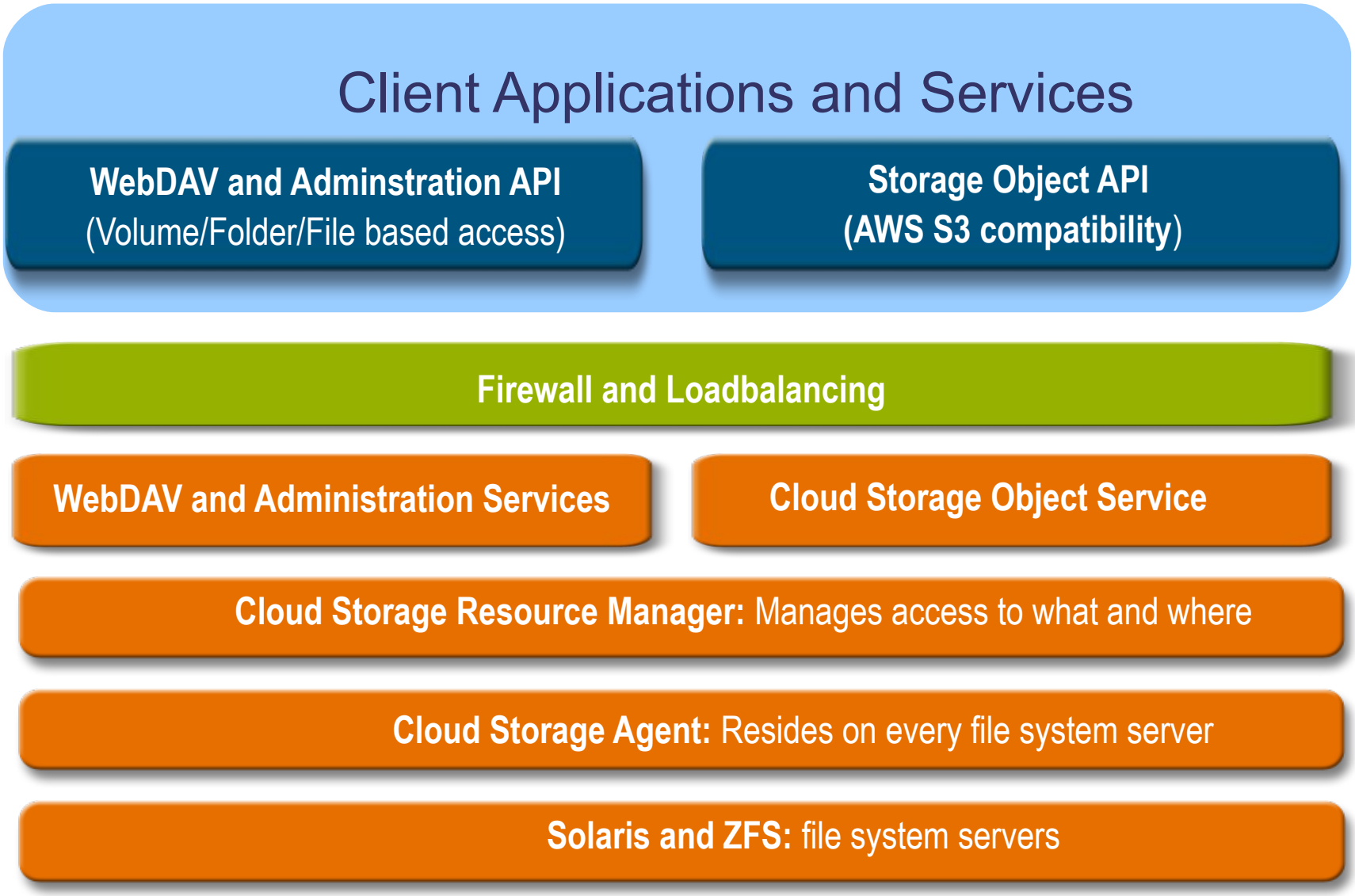




/home808wooster/ (bucket)

- /object 1
- /object 2
- /object 3

Storage Service Architecture



Get Started Today

- Your IT will not change into Cloud overnight
- But you can start small and grow!
- Try out some pilot projects
- Gain experience
- Sign up for Early Access to Sun Cloud Services
- Let Sun experts help you take advantage of Cloud Computing with Assessment Services and Workshops
  - > Gather business requirements
  - > Create a high-level architecture
  - > Industry-best practices and proven methodologies to create value-added solutions

http://sun.com/cloud