ITERATIVE MAPREDUCE

WHAT IS ITERATIVE MAPREDUCE

- Iterative MapReduce
  - Mapreduce is a Programming Model instantiating the paradigm of bringing computation to data
  - Iterative Mapreduce extends Mapreduce programming model and support iterative algorithms for Data Mining and Data Analysis
- Interoperability
  - Is it possible to use the same computational tools on HPC and Cloud?
  - Enabling scientists to focus on science not programming distributed systems
- Reproducibility
  - Is it possible to use Cloud Computing for Scalable, Reproducible Experimentation?
  - Sharing results, data, and software

TWISTER ITERATIVE MAPREDUCE V0.9

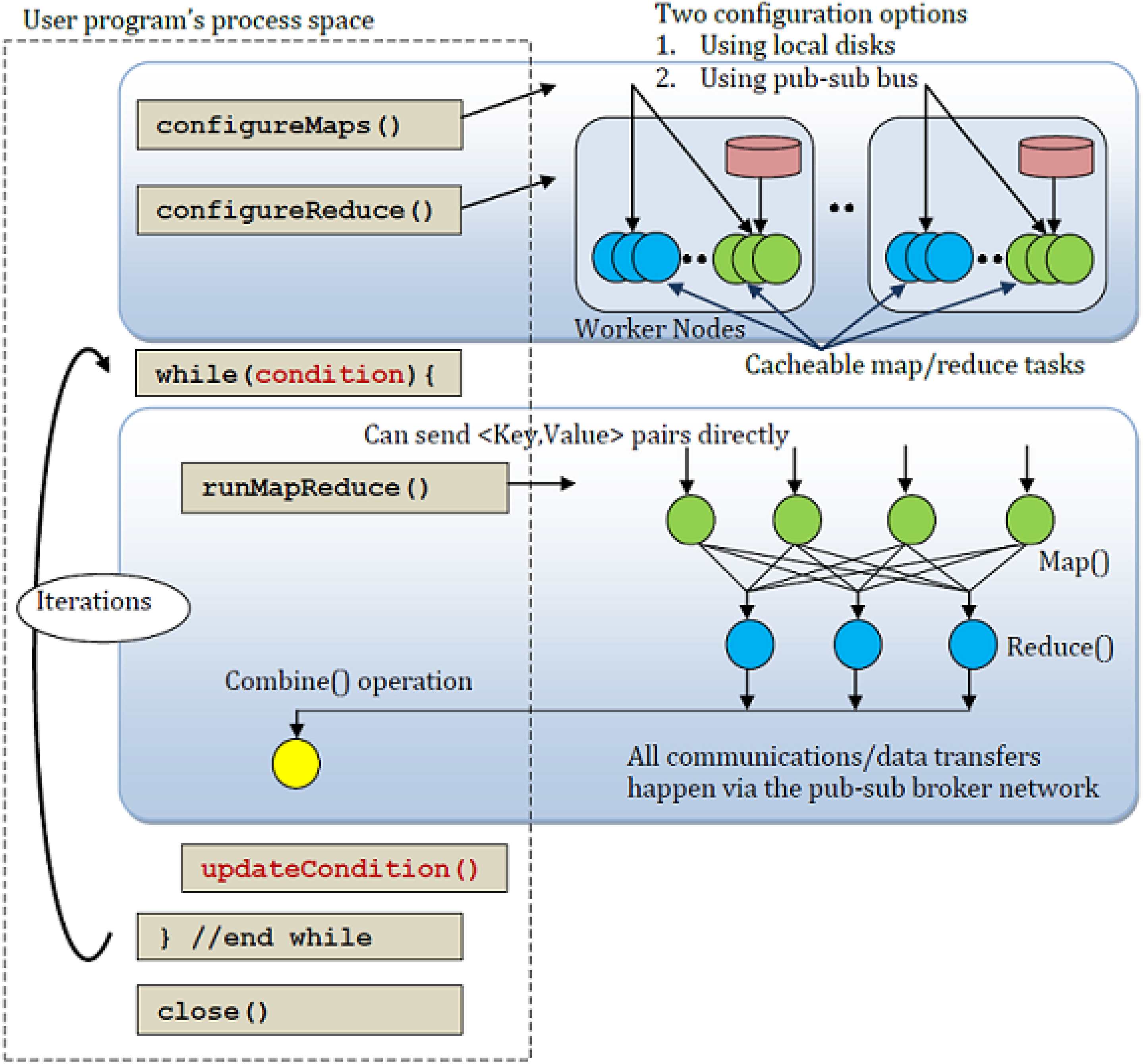
- Distinction on static and variable data
- Configurable long running (cacheable) map/reduce tasks
- Pub/sub messaging based communication/data transfers
- Broker Network for facilitating communication
  - Version 0.9 (hands-on)
  - Version beta

ITERATIVE MAPREDUCE FRAMEWORKS

- Twister
  - Map->Reduce->Combine->Broadcast
  - Long running map tasks (data in memory)
  - Centralized driver based, statically scheduled.
- Daytona
  - Microsoft Iterative MapReduce on Azure using cloud services
  - Architecture similar to Twister
- Twister4Azure
  - A decentralized Map/reduce input caching, reduce output caching
- Haloop
  - On disk caching, Map/reduce input caching, reduce output caching
- Spark
  - Iterative Mapreduce Using Resilient Distributed Dataset to ensure the fault tolerance
- Pregel
  - Graph processing from Google
- Mahout
  - Apache project for supporting data mining algorithms

ITERATIVE MAPREDUCE FRAMEWORKS

- Network Levitated Merge
    - RDMA/infiniband based shuffle & merge
- Mate-EC2
    - Local reduction object
- Asynchronous Algorithms in MapReduce
    - Local & global reduce
- MapReduce online
    - online aggregation, and continuous queries
    - Push data from Map to Reduce
- Orchestra
    - Data transfer improvements for MR
- iMapReduce
    - Async iterations, One to one map & reduce mapping, automatically joins loop-variant and invariant data
- CloudMapReduce & Google AppEngine MapReduce
    - MapReduce frameworks utilizing cloud infrastructure services



User program's process space

Two configuration options

1. Using local disks
2. Using pub-sub bus

configureMaps()

configureReduce()

Worker Nodes

Cacheable map/reduce tasks

while(condition){

Can send <Key,Value> pairs directly

runMapReduce()

Map()

Iterations

Reduce()

Combine() operation

All communications/data transfers happen via the pub-sub broker network

updateCondition()

} //end while

close()

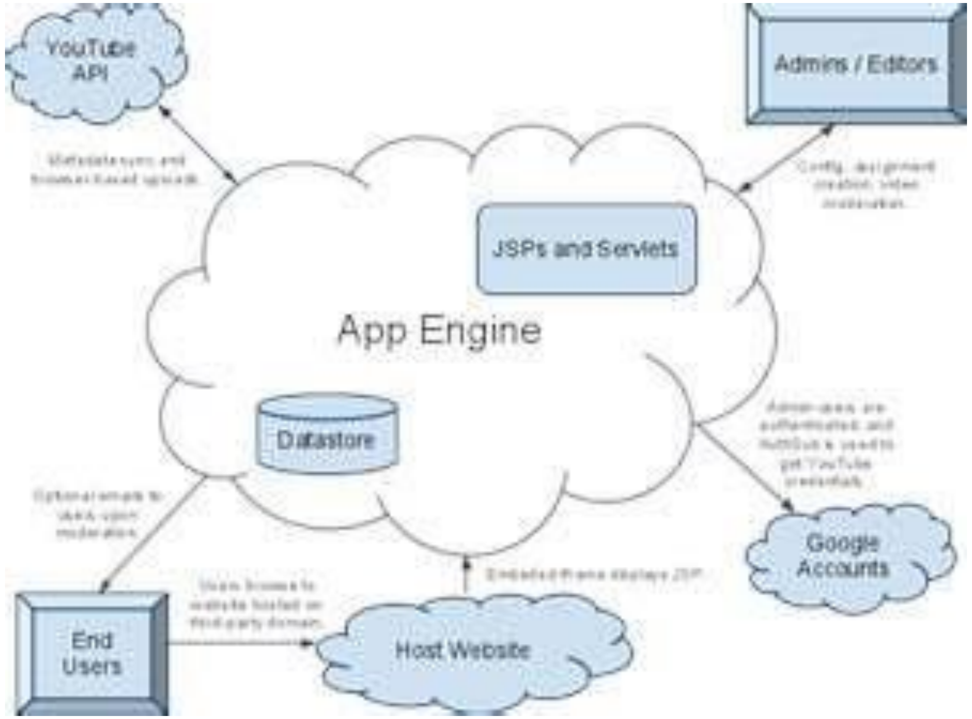
TWISTER

- http://www.iterativemapreduce.org/

INTRODUCTION TO GOOGLE APP ENGINE

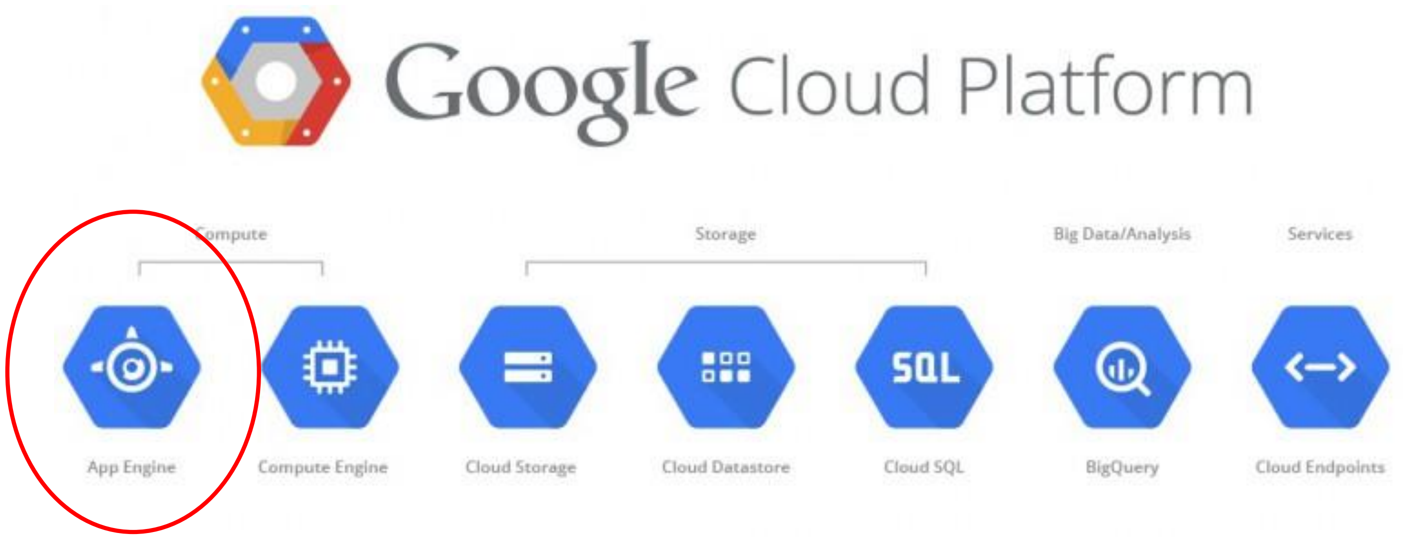
GOOGLE APP ENGINE

- Does one thing well: running web apps
- Simple app configuration
- Scalable
- Secure

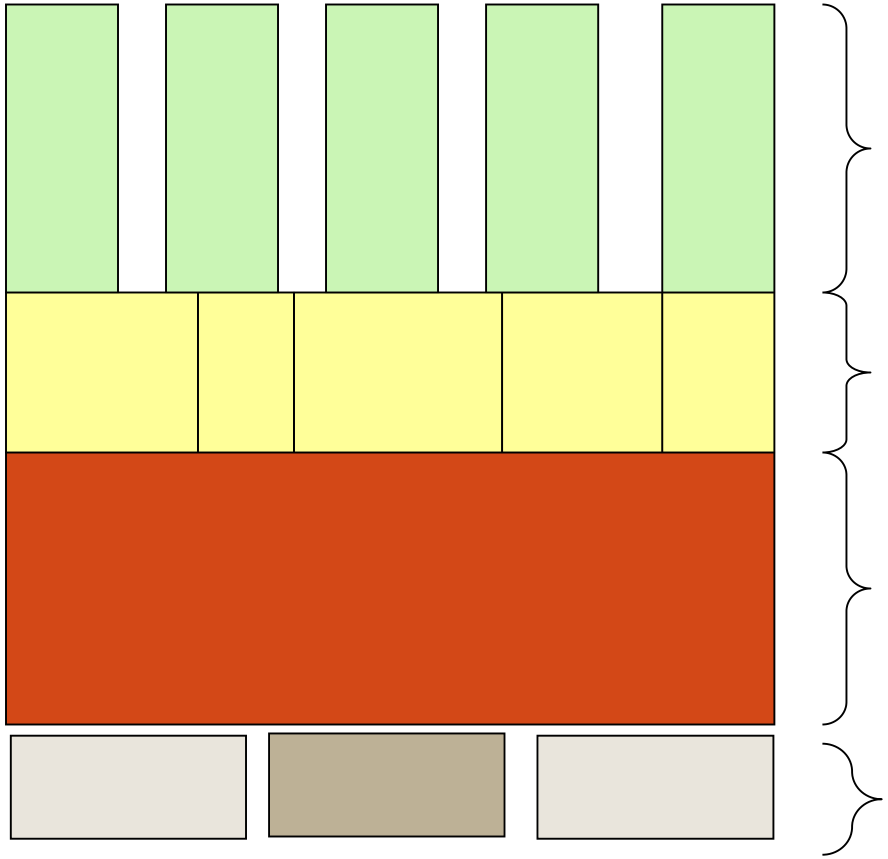


GAE is part of Google Cloud and is Platform as a Service Cloud (PaaS)

- Using Googles Infrastructure to host and build your Web Applications
- Free Account --- for limited bandwidth ….see http://code.google.com/appengine/ for details



INFRASTRUCTURE VS. PLATFORM - WHAT IS "THE PLATFORM"?



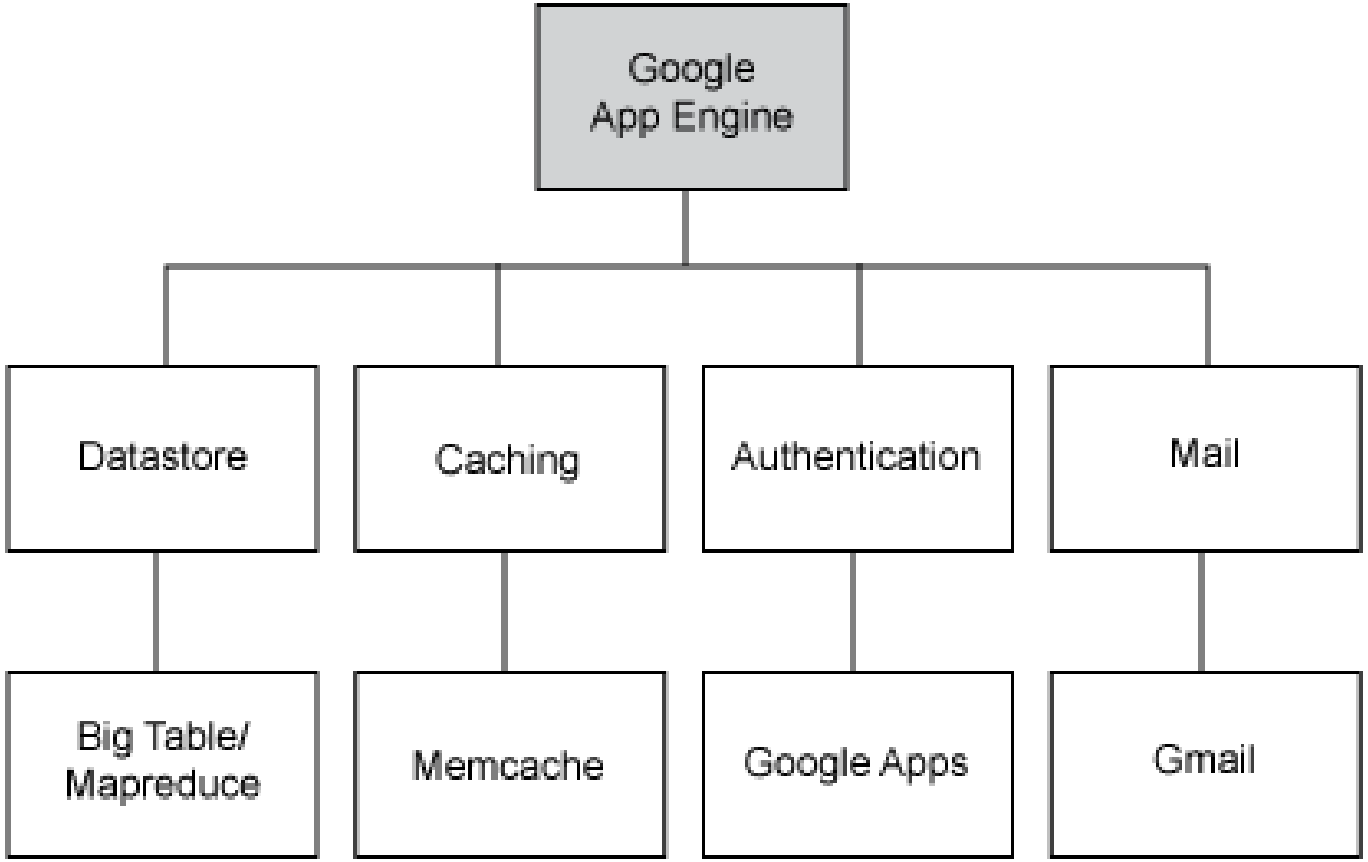
Application-specific code

Libraries: shared by multiple applications

Platform: same for all applications

infrastructure: hidden by platform

WHAT DOES GAE PROVIDE



GAE PROVIDES ---WHY NOT ON YOUR OWN?

- how many developers or architects have experience and the mindset to build applications that support 100s of thousands of concurrent users up all the time?
- Scaling Big is Really Hard
- Horizontal scaling model
  - this is not a model that most web application developers have experience with.
  - instead of using more capable hardware (vertical scaling), you use more instances of less-capable hardware, each handling a slice of the work, often doing the same function (e.g. sliced between groups of users).
  - intent is to reduce centralization of resources
  - ultimate goal is to simply be able to add more instances of the hardware *without limit* to meet increased scale requirements.

APP ENGINE DOES ONE THING WELL

- App Engine handles HTTP(S) requests, nothing else
  - Think RPC: request in, processing, response out
  - Works well for the web and AJAX; also for other services

- App configuration is simple
  - No performance tuning needed

- Everything is built to scale
  - “infinite” number of apps, requests/sec, storage capacity
  - APIs are simple

GAE HAS LIMITATIONS WITH FREE ACCOUNT

- What is Free and What is NOT
- **FREE:** All applications have a default quota configuration, the "free quotas", which should allow for roughly 5 million page views a month for an efficient application. You can read more about system quotas in the quota documentation.
- **PAY FOR MORE:** As your application grows, it may need a higher resource allocation than the default quota configuration provides. You can purchase additional computing resources by enabling billing for your application. Billing enables developers to raise the limits on all system resources and pay for even higher limits on CPU, bandwidth, storage, and email usage

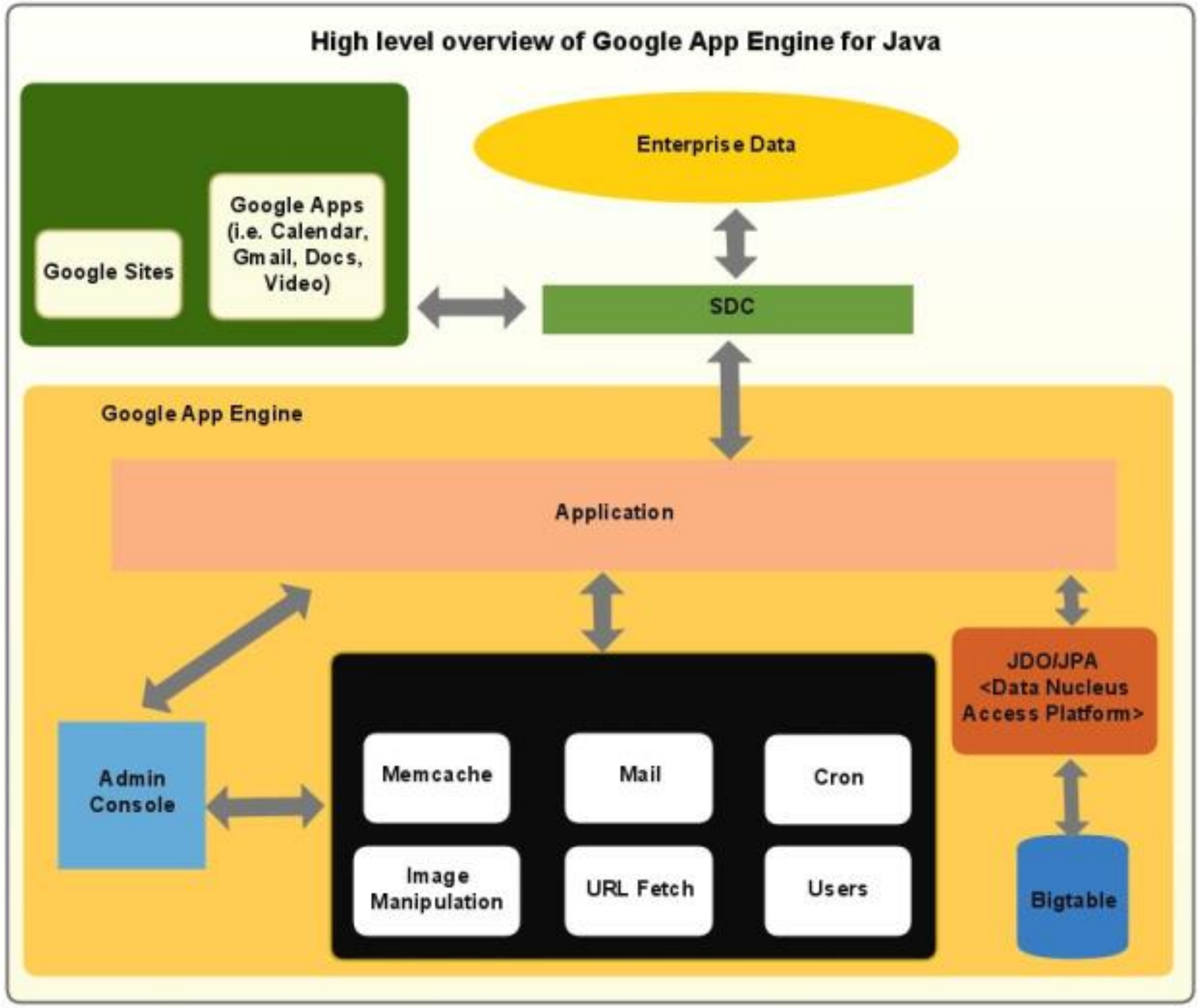
SERVICES

- URLFetch – fetch web resources/services
- Images – manipulate images: resize, rotate, flip, crop
- Google Accounts
- Mail
- XMPP – instant messages
- Task Queue – message queue; allow integration with non-GAPPs
- Datastore – managing data objects
- Blobstore – large files, much larger than objects in datastore, use <key, object> to access

WHAT KIND OF "APPS" CAN GOOGLE APP ENGINE SUPPORT

- What languages are supported: **python, java, php and Go**

APP ENGINE ARCHITECTURE (JAVA)

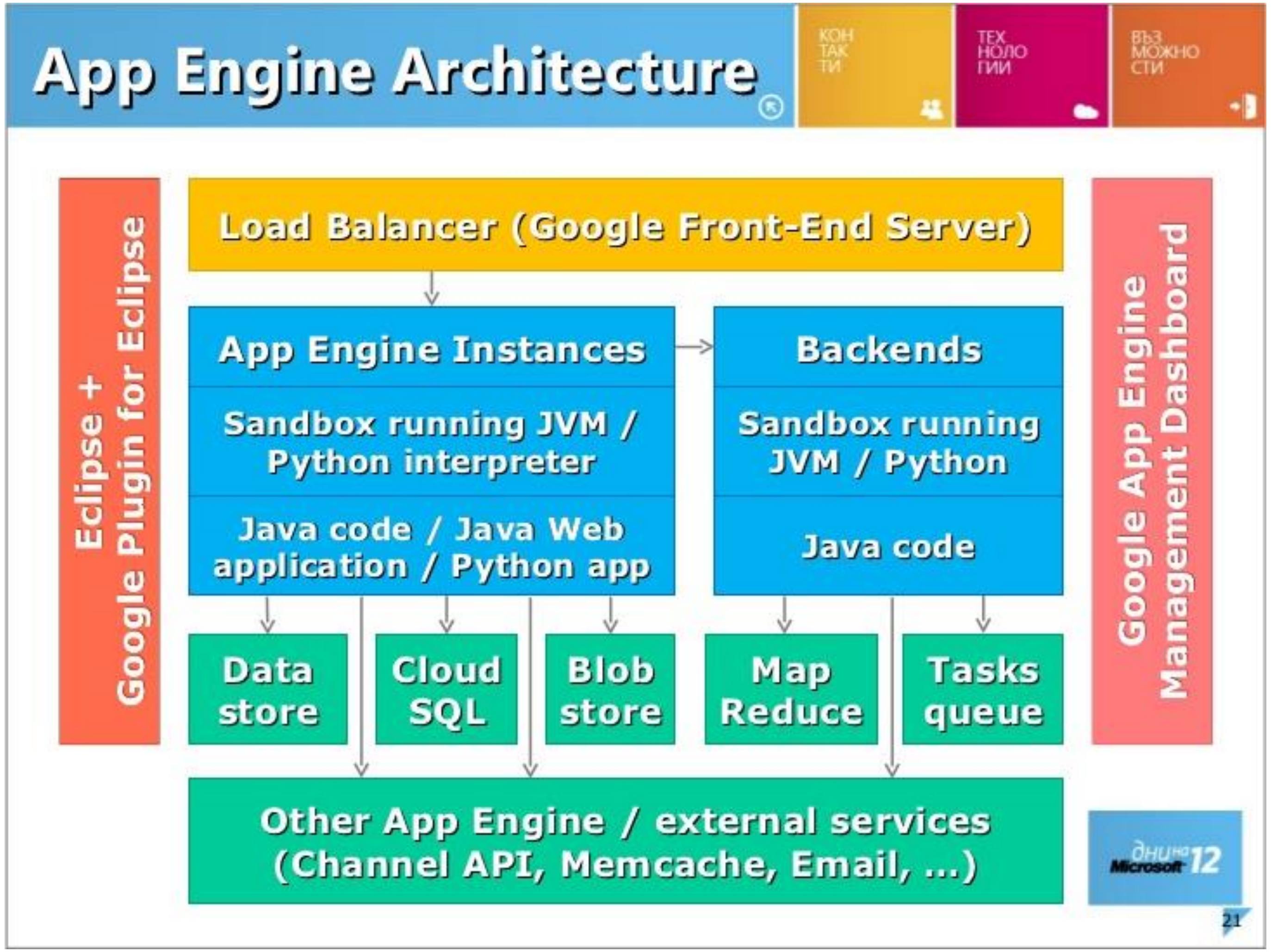


High level overview of Google App Engine for Java
Enterprise Data
Google Sites
Google Apps (i.e. Calendar, Gmail, Docs, Video)
SDC
Google App Engine
Application
Admin Console
Memcache
Mail
Cron
Image Manipulation
URL Fetch
Users
JDO/JPA <Data Nucleus Access Platform>
Bigtable

SDC:
Secure data connector

JDO: java data object
JPA: java persistent API

GAE ARCHITECTURE FEATURING JAVA



App Engine Architecture

КОН ТАК ТИ

ТЕХ НОЛО ГИИ

ВЪЗ МОЖНО СТИ

Eclipse + Google Plugin for Eclipse

Load Balancer (Google Front-End Server)

App Engine Instances

Sandbox running JVM / Python interpreter

Java code / Java Web application / Python app

Backends

Sandbox running JVM / Python

Java code

Data store

Cloud SQL

Blob store

Map Reduce

Tasks queue

Other App Engine / external services (Channel API, Memcache, Email, ...)

Google App Engine Management Dashboard

GAE: JAVA OR PYTHON? OR HOW ABOUT PHP OR GO

- First Question: What do you or your developers know?
- Benefit of Python: powerful python syntax, library, possibly shorter code
- Benefit of Java: rich language that is mature, many packages, can use JDO/JPA
  - Better portability if you need to use Bigtable to store data

JAVA ON GAE

- Java Servlets, JSPs on Google AppEngine
- **You provide your app's servlet classes, JavaServer Pages (JSPs), static files and data files, along with the deployment descriptor (the web.xml file) and other configuration files, in a standard WAR directory structure.**
- **App Engine serves requests by invoking servlets according to the deployment descriptor.**

SCALING AND RESULTING APP LIMITATIONS

SCALING IN GAE HOW IS IT ACHIEVED --- LEADS US TO SOME LIMITATIONS IN CODING

- Low-usage apps: many apps per physical host
- High-usage apps: multiple physical hosts per app

- Stateless APIs are trivial to replicate

- Datastore built on top of Bigtable; designed to scale well
  - Abstraction *on top of* Bigtable
  - API influenced by scalability
    - No joins
    - Recommendations: *denormalize* schema; precompute joins

RESTRICTIONS ON JAVA IN GAE – A NEW WAY OF THINKING TO GET SCALABILITY

- To allow App Engine to distribute requests for applications across multiple web servers, and to prevent one application from interfering with another, the application runs in a restricted "sandbox" environment.
- The JVM runs in a secured "sandbox" environment to isolate your application for service and security.
- The sandbox ensures that apps can only perform actions that do not interfere with the performance and scalability of other apps.

RESTRICTIONS ON JAVA IN GAE – A NEW WAY OF THINKING TO GET SCALABILITY

- **An app cannot spawn threads, write data to the local file system or make arbitrary network connections.** **(cant allow to store to local file system when things are distributed ---could be problems…you need to use the datastore instead)**
  - **HOWEVER--Apps use the URL Fetch service to access resources over the web, and to communicate with other hosts using the HTTP and HTTPS protocols. Java apps can simply usejava.net.URLConnection and related classes from the Java standard library to access this service.**

- **app also cannot use JNI or other native code.**

MORE ON SCALING

AUTOMATIC SCALING TO APPLICATION NEEDS

- You don’t need to configure your resource needs
- One CPU can handle many requests per second
- Apps are hashed onto CPUs:
  - One process per app, many apps per CPU
  - Creating a new process is a matter of cloning a generic “model” process and then loading the application code (in fact the clones are pre-created and sit in a queue)
  - The process hangs around to handle more requests (reuse)
  - Eventually old processes are killed (recycle)
- Busy apps (many QPS query per sec) get assigned to multiple CPUs
  - This automatically adapts to the need
    - as long as CPUs are available

PRESERVING FAIRNESS THROUGH QUOTAS

- Everything an app does is limited by quotas, for example:
  - request count, bandwidth used, CPU usage, datastore call count, disk space used, emails sent, even errors!
- If you run out of quota that particular operation is blocked (raising an exception) for a while (~10 min) until replenished
- Free quotas are tuned so that a well-written app (light CPU/datastore use) can survive a moderate "slashdotting"
- **The point of quotas is to be able to support a very large number of small apps (analogy: baggage limit in air travel)**
- Large apps need raised quotas
  - currently this is a manual process (search FAQ for "quota")
  - in the future you can buy more resources

WHAT ABOUT DATA

GAE DATASTORE (STORAGE ORGANIZATION)

- Data model
  - Property, entity, entity group
  - Schemeless: properties can have different types/meanings for different objects
  - Allow (1) object query (2) SQL-like query
- Transaction
- Can be applied to a group of operations
- Persistent store (check BigTable paper)
  - Strongly consistent
  - Not relational database
  - Index built-in
- Memcache
  - Caches objects from bigtable, to improve performance

HIERARCHICAL DATASTORE

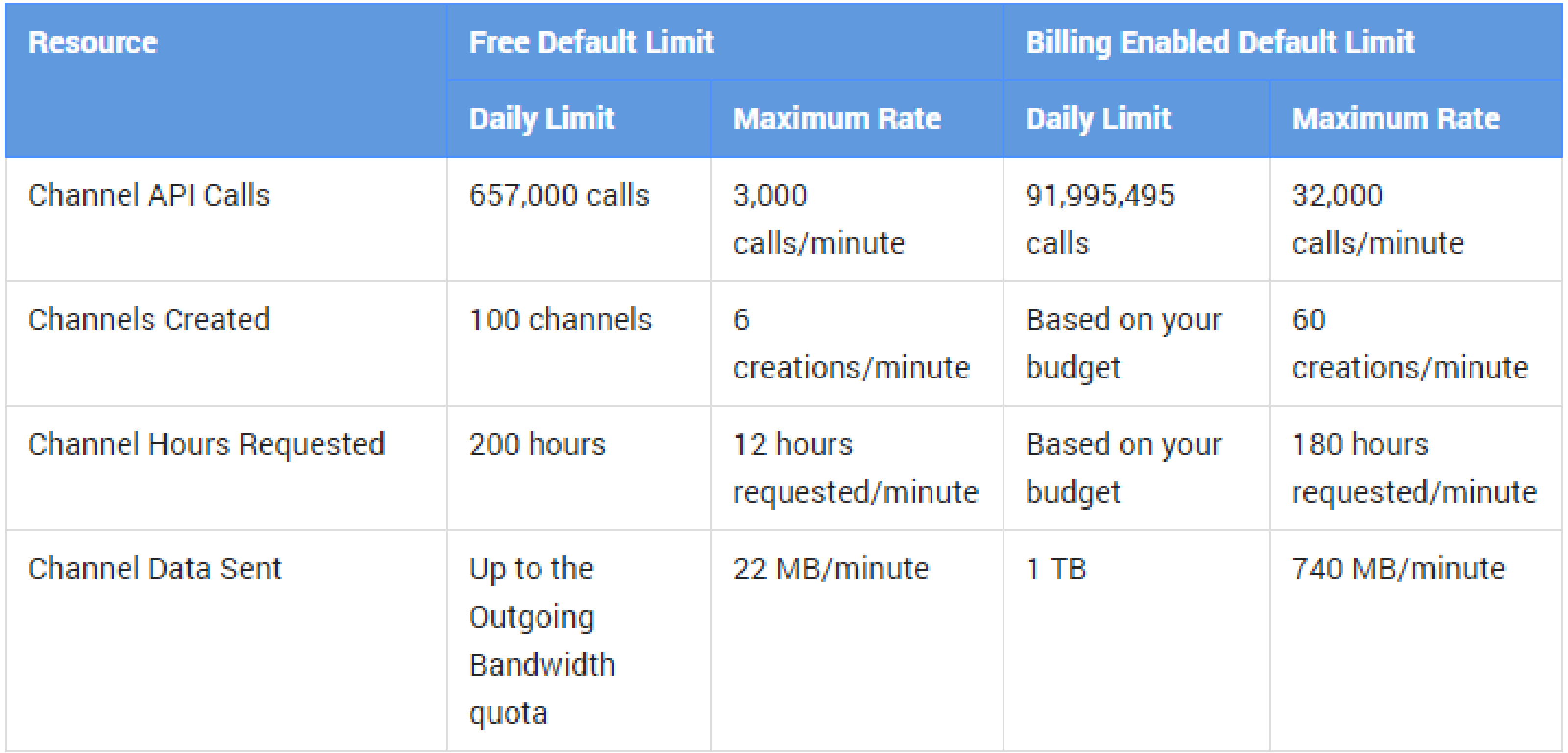
- *Entities* have a *Kind,* a *Key,* and *Properties*
  - Entity ~~ Record ~~ Python dict ~~ Python class instance
  - Key ~~ structured foreign key; includes Kind
  - Kind ~~ Table ~~ Python class
  - Property ~~ Column or Field; has a type
- Dynamically typed: Property types are recorded per Entity
- Key has either *id* or *name*
  - the id is auto-assigned; alternatively, the name is set by app
  - A key can be a *path* including the parent key, and so on
- Paths define *entity groups* which limit *transactions*
  - A transaction locks the *root entity* (parentless ancestor key)
  - Recall the chubby lock service in bigtable paper

INDEXES

- Properties are automatically indexed by type+value
  - There is an index for each Kind / property name combo
  - Whenever an entity is written all relevant indexes are updated
  - However Blob and Text properties are never indexed
- This supports basic queries: AND on property equality
- For more advanced query needs, create *composite indexes*
  - SDK auto-updates index.yaml based on queries executed
  - These support inequalities (<, <=, >, >=) and result ordering
  - Index building has to scan *all* entities due to parent keys

FREE TIER

- First 5GB
- Daily limits see Online for CURRENT QUOTAS



Resource	Free Default Limit		Billing Enabled Default Limit	
	Daily Limit	Maximum Rate	Daily Limit	Maximum Rate
Channel API Calls	657,000 calls	3,000 calls/minute	91,995,495 calls	32,000 calls/minute
Channels Created	100 channels	6 creations/minute	Based on your budget	60 creations/minute
Channel Hours Requested	200 hours	12 hours requested/minute	Based on your budget	180 hours requested/minute
Channel Data Sent	Up to the Outgoing Bandwidth quota	22 MB/minute	1 TB	740 MB/minute

WHAT HAPPENS WHEN YOU EXCEED YOUR BUDGET (FREE OR WHAT YOU SET)

- **When an application consumes all of an allocated resource, the resource becomes unavailable until the quota is replenished.**
- **This may mean that your application will not work until the quota is replenished.**

WHAT HAPPENS WHEN YOU EXCEED YOUR BUDGET (FREE OR WHAT YOU SET)

- **For resources that are required to initiate a request, when the resource is depleted, App Engine by default returns an HTTP 403 Forbidden status code for the request instead of calling a request handler. The following resources have this behavior:**
  - **Bandwidth, incoming and outgoing**

- **For all other resources, when the resource is depleted, an attempt in the application to consume the resource results in an exception. This exception can be caught by the application and handled, such as by displaying a friendly error message to the user. In the Python API, this exception is apiproxy_errors.OverQuotaError. In the Java API, this exception is com.google.apphosting.api.ApiProxy.OverQuotaException.**

SECURITY

- Constrain direct OS functionality
  - no processes, threads, dynamic library loading
  - no sockets (use urlfetch API instead)
  - can't write files (use datastore)
  - disallow unsafe Python extensions (e.g. ctypes)
- Limit resource usage
  - Hard time limit of 30 seconds per request
  - Most requests must use less than 300 msec CPU time
  - Hard limit of 1MB on request/response size, API call size, etc.
  - Quota system for number of requests, API calls, emails sent, etc
  - Free use for 500MB data and 5M requests per month
  - 10 applications per account

Amazon Web Services

WHAT IS AMAZON WEB SERVICES ?

- Amazon Web Services (AWS) is a collection of remote computing services (web services) that together make up a cloud computing platform, offered over the Internet by Amazon.com.
- Website: https://aws.amazon.com/

WHAT IS AWS OFFERING?

- **Low Ongoing Cost:,** pay-as-you-go pricing with no up-front expenses or long-term commitments.
- **Instant Elasticity & Flexible Capacity:** (scaling up and down) Eliminate guessing on your infrastructure capacity needs.
- **Speed & Agility:** Develop and deploy applications faster Instead of waiting weeks or months for hardware to arrive and get installed.
- **Apps not Ops:** Focus on projects. Lets you shift resources away from data center investments and operations and move them to innovative new projects.
- **Global Reach:** Take your apps global in minutes.
- **Open and Flexible:** You choose the development platform or programming model that makes the most sense for your business.
- **Secure:** Allows your application to take advantage of the multiple layers of operational and physical security in the AWS data centers to ensure the integrity and safety of your data.

AMAZON WEB SERVICES CLOUD PLATFORM

- AWS consists of many cloud services that you can use in combinations tailored to your business or organizational needs.
- This section introduces the major AWS services by category. To access the services, you can use the
  - AWS Management Console,
  - The Command Line Interface,
  - Software Development Kits (SDKs).

AWS MANAGEMENT CONSOLE

- Access and manage Amazon Web Services through the AWS Management Console, a simple and intuitive user interface.
- You can also use the AWS Console Mobile Application to quickly view resources on the go.
- The AWS Management Console brings the unmatched breadth and depth of AWS right to your computer or mobile phone with a secure, easy-to-access, web-based portal.
- Discover new services, manage your entire account, build new applications, and learn how to do even more with AWS.

For Reference: https://aws.amazon.com/console/

AWS COMMAND LINE INTERFACE

- The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services.
- With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.
- For reference: https://aws.amazon.com/cli/

SOFTWARE DEVELOPMENT KITS

- Software Development Kits (SDKs) simplify using AWS services in your applications with an Application Program Interface (API) tailored to your programming language or platform.
- For reference: https://aws.amazon.com/tools/

AVAILABLE SERVICES MAINLY ON:

- Analytics
- Application Integration
- AR and VR
- AWS Cost Management
- Blockchain
- Business Applications
- Compute
- Customer Engagement
- Database
- Developer Tools
- Game Tech
- Internet of Things (IoT)
- Machine Learning
- Security, Identity, and Compliance

ANALYTICS

- **Amazon Athena** - Query data in S3 using SQL
- **Amazon CloudSearch** - Managed search service
- **Amazon Elasticsearch Service** - Run and scale Elasticsearch clusters
- **Amazon EMR** - Hosted Hadoop framework
- **Amazon Kinesis** - Analyze real-time video and data streams
- **Amazon Managed Streaming for Apache Kafka** - Fully managed Apache Kafka service
- **Amazon Redshift** - Fast, simple, cost-effective data warehousing
- **Amazon QuickSight** - Fast business analytics service
- **AWS Data Exchange** - Find, subscribe to, and use third-party data in the cloud
- **AWS Data Pipeline** - Orchestration service for periodic, data-driven workflows
- **AWS Glue** - Prepare and load data
- **AWS Lake Formation** - Build a secure data lake in days

APPLICATION INTEGRATION

- **AWS Step Functions** - Coordination for distributed applications
- **Amazon AppFlow** - No code integration for SaaS apps & AWS services
- **Amazon EventBridge** - Serverless event bus for SaaS apps & AWS services
- **Amazon MQ** - Managed message broker for ActiveMQ
- **Amazon Simple Notification Service (SNS)** - Pub/sub, SMS, email, and mobile push notifications
- **Amazon Simple Queue Service (SQS)** - Managed message queues
- **AWS AppSync** - Power your apps with the right data from many sources, at scale

AR AND VR

- **Amazon Sumerian** - Build and run VR and AR applications
- Amazon Sumerian lets you create and run virtual reality (VR), augmented reality (AR), and 3D applications quickly and easily without requiring any specialized programming or 3D graphics expertise.
- With Sumerian, you can build highly immersive and interactive scenes that run on popular hardware such as Oculus Go, Oculus Rift, HTC Vive, HTC Vive Pro, Google Daydream, and Lenovo Mirage as well as Android and iOS mobile devices.

AWS COST MANAGEMENT

- **AWS Cost Explorer** - Analyze your AWS cost and usage
- **AWS Budgets** - Set custom cost and usage budgets
- **AWS Cost and Usage Report** - Access comprehensive cost and usage information
- **Reserved Instance Reporting** - Dive deeper into your reserved instances (RIs)

BLOCKCHAIN

- **Amazon Managed Blockchain** - Fully managed service that makes it easy to create and manage scalable blockchain networks using the popular open source frameworks Hyperledger Fabric and Ethereum.
- **Amazon Quantum Ledger Database (QLDB)** - Fully managed ledger database

BUSINESS APPLICATIONS

- **Alexa for Business** - Empower your organization with Alexa
- **Amazon Chime** - Frustration-free meetings, video calls, and chat
- **Amazon Honeycode (Beta)** - Build mobile & web apps without programming
- **Amazon WorkDocs** - Secure enterprise document storage and sharing
- **Amazon WorkMail** - Secure email and calendaring

COMPUTE

- **Amazon EC2** - Virtual servers in the cloud
- **Amazon EC2 Auto Scaling** - Scale compute capacity to meet demand
- **Amazon Lightsail** - Launch and manage virtual private servers
- **AWS Batch** - Run batch jobs at any scale
- **AWS Elastic Beanstalk** - Run and manage web apps
- **AWS Lambda** - Run code without thinking about servers
- **AWS Outposts** - Run AWS infrastructure on-premises
- **AWS Serverless Application Repository** - Discover, deploy, and publish serverless applications
- **AWS Snow Family** - Physical devices to aggregate and process data in edge locations, then transfer to AWS
- **AWS Wavelength** - Deliver ultra-low latency applications for 5G devices
- **VMware Cloud on AWS** - Build a hybrid cloud without custom hardware

CUSTOMER ENGAGEMENT

- Amazon Connect - Cloud-based contact center service
- Amazon Pinpoint - Personalized user engagement across channels
- Amazon Simple Email Service (SES) - Email sending and receiving
- Contact Lens for Amazon Connect - Contact center analytics powered by ML

DATABASE

- Amazon Aurora - High performance managed relational database
- Amazon DynamoDB - Managed NoSQL database
- Amazon DocumentDB (with MongoDB compatibility) - Fully managed document database
- Amazon ElastiCache - In-memory caching system
- Amazon Keyspaces (for Apache Cassandra) - Managed Cassandra-compatible database
- Amazon Neptune - Fully managed graph database service
- Amazon Quantum Ledger Database (QLDB) - Fully managed ledger database
- Amazon RDS - Managed relational database service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB
- Amazon RDS on VMware - Automate on-premises database management
- Amazon Redshift - Fast, simple, cost-effective data warehousing
- Amazon Timestream - Fully managed time series database
- AWS Database Migration Service - Migrate databases with minimal downtime

DEVELOPER TOOLS

- Amazon Corretto - Production-ready distribution of OpenJDK
- AWS Cloud Development Kit (CDK) - Model cloud infrastructure using code
- AWS Cloud9 - Write, run, and debug code on a cloud IDE
- AWS CodeArtifact - Secure, scalable, and cost-effective artifact management for software development
- AWS CodeBuild - Build and test code
- AWS CodeCommit - Store code in private Git repositories
- AWS CodeDeploy - Automate code deployment
- Amazon CodeGuru - Find your most expensive lines of code
- AWS CodePipeline - Release software using continuous delivery
- AWS CodeStar - Develop and deploy AWS applications
- AWS Command Line Interface - Unified tool to manage AWS services
- AWS Device Farm - Test Android, iOS, and web apps on real devices in the AWS cloud
- AWS X-Ray - Analyze and debug your applications

GAME TECH

- Amazon GameLift - Simple, fast, cost-effective dedicated game server hosting
- Amazon Lumberyard - A free cross-platform 3D game engine, with Full Source, integrated with AWS and Twitch

INTERNET OF THINGS (IOT)

- AWS IoT Core - Connect devices to the cloud
- AWS Greengrass - Local compute, messaging, and sync for devices
- AWS IoT 1-Click - One click creation of an AWS Lambda trigger
- AWS IoT Analytics - Analytics for IoT devices
- AWS IoT Button - Cloud programmable dash button
- AWS IoT Device Defender - Security management for IoT devices
- AWS IoT Device Management - Onboard, organize, and remotely manage IoT devices
- AWS IoT Events - IoT event detection and response
- AWS IoT SiteWise - IoT data collector and interpreter
- AWS IoT Things Graph - Easily connect devices and web services
- AWS Partner Device Catalog - Curated catalog of AWS-compatible IoT hardware
- FreeRTOS - Real-time operating system for microcontrollers

MACHINE LEARNING

- Amazon SageMaker - Build, train, and deploy machine learning models at scale
- Amazon Augmented AI - Easily implement human review of ML predictions
- Amazon CodeGuru - Find your most expensive lines of code
- Amazon Comprehend - Discover insights and relationships in text
- Amazon Elastic Inference - Deep learning inference acceleration
- Amazon Forecast - Increase forecast accuracy using machine learning
- Amazon Fraud Detector - Detect more online fraud faster
- Amazon Kendra - Reinvent enterprise search with ML
- Amazon Lex - Build voice and text chatbots
- Amazon Personalize - Build real-time recommendations into your applications
- Amazon Polly - Turn text into life-like speech

- Amazon Rekognition - Analyze image and video
- Amazon SageMaker Ground Truth - Build accurate ML training datasets
- Amazon Textract - Extract text and data from documents
- Amazon Translate - Natural and fluent language translation
- Amazon Transcribe - Automatic speech recognition
- AWS Deep Learning AMIs - Deep learning on Amazon EC2
- AWS Deep Learning Containers - Docker images for deep learning
- AWS DeepComposer - ML enabled musical keyboard
- AWS DeepLens - Deep learning enabled video camera
- AWS Inferentia - Machine learning inference chip
- Apache MXNet on AWS - Scalable, open-source deep learning framework
- PyTorch on AWS - Flexible open-source machine learning framework
- TensorFlow on AWS - Open-source machine intelligence library

SECURITY, IDENTITY, AND COMPLIANCE

- AWS Identity and Access Management (IAM) - Securely manage access to services and resources
- Amazon Cognito - Identity management for your apps
- Amazon Detective - Investigate potential security issues
- Amazon GuardDuty - Managed threat detection service
- Amazon Inspector - Analyze application security
- Amazon Macie - Discover and protect your sensitive data at scale
- AWS Artifact - On-demand access to AWS' compliance reports
- AWS Certificate Manager - Provision, manage, and deploy SSL/TLS certificates
- AWS CloudHSM - Hardware-based key storage for regulatory compliance

- AWS Directory Service - Host and manage active directory
- AWS Firewall Manager - Central management of firewall rules
- AWS Key Management Service - Managed creation and control of encryption keys
- AWS Resource Access Manager - Simple, secure service to share AWS resources
- AWS Secrets Manager - Rotate, manage, and retrieve secrets
- AWS Security Hub - Unified security and compliance center
- AWS Shield - DDoS protection
- AWS Single Sign-On - Cloud single sign-on (SSO) service
- AWS WAF - Filter malicious web traffic

STORAGE

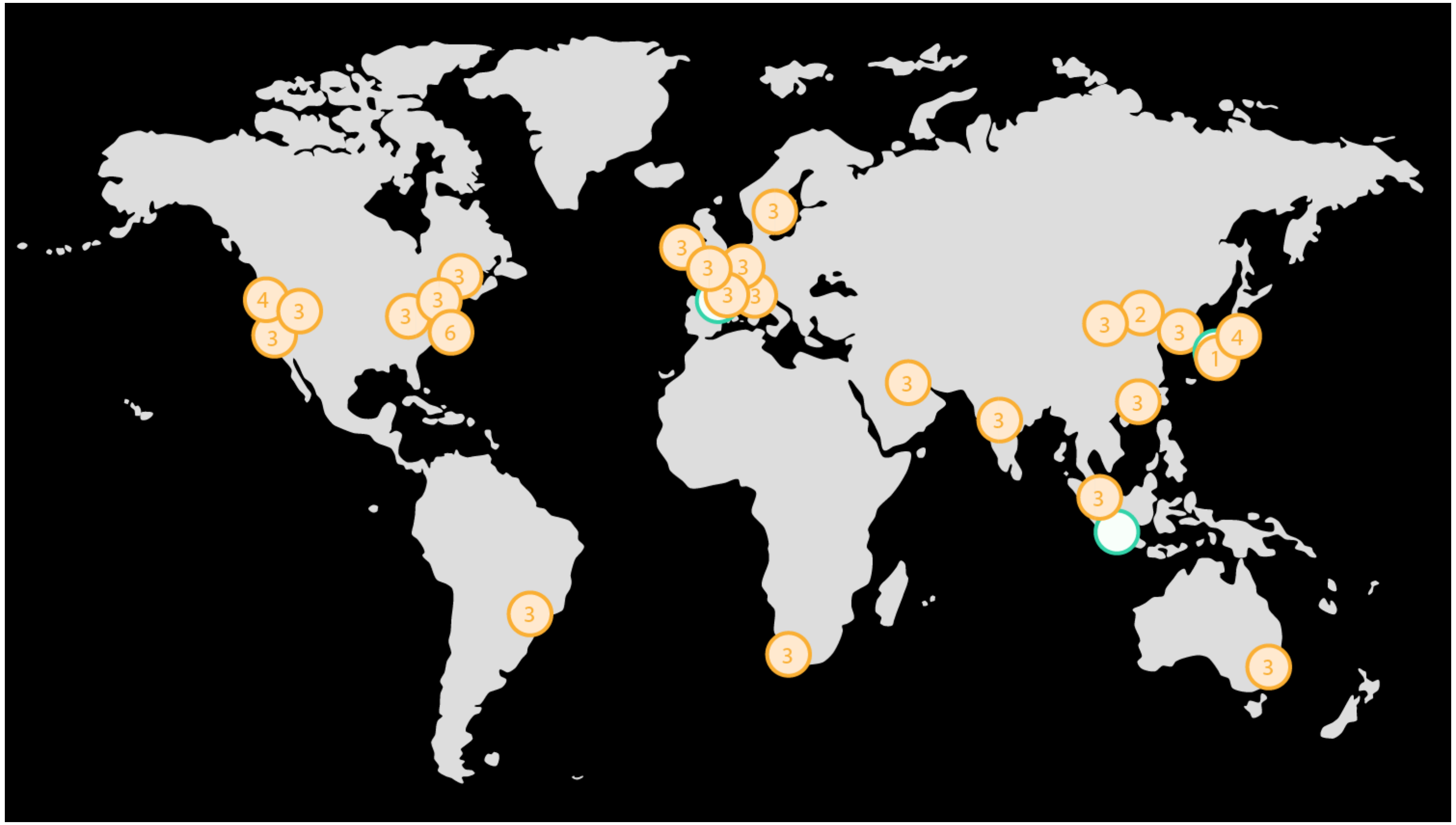
- Amazon Simple Storage Service (S3) - Scalable storage in the cloud
- Amazon Elastic Block Store (EBS) - EC2 block storage volumes
- Amazon Elastic File System (EFS) - Fully managed file system for EC2
- Amazon FSx for Windows File Server - Fully managed Windows native file system
- Amazon S3 Glacier - Low-cost archive storage in the cloud
- AWS Backup - Centralized backup across AWS services
- AWS Snow Family - Physical edge computing and storage devices for rugged or disconnected environments
- AWS Storage Gateway - Hybrid storage integration
- CloudEndure Disaster Recovery - Highly automated disaster recovery

FOR DOCUMENTATION

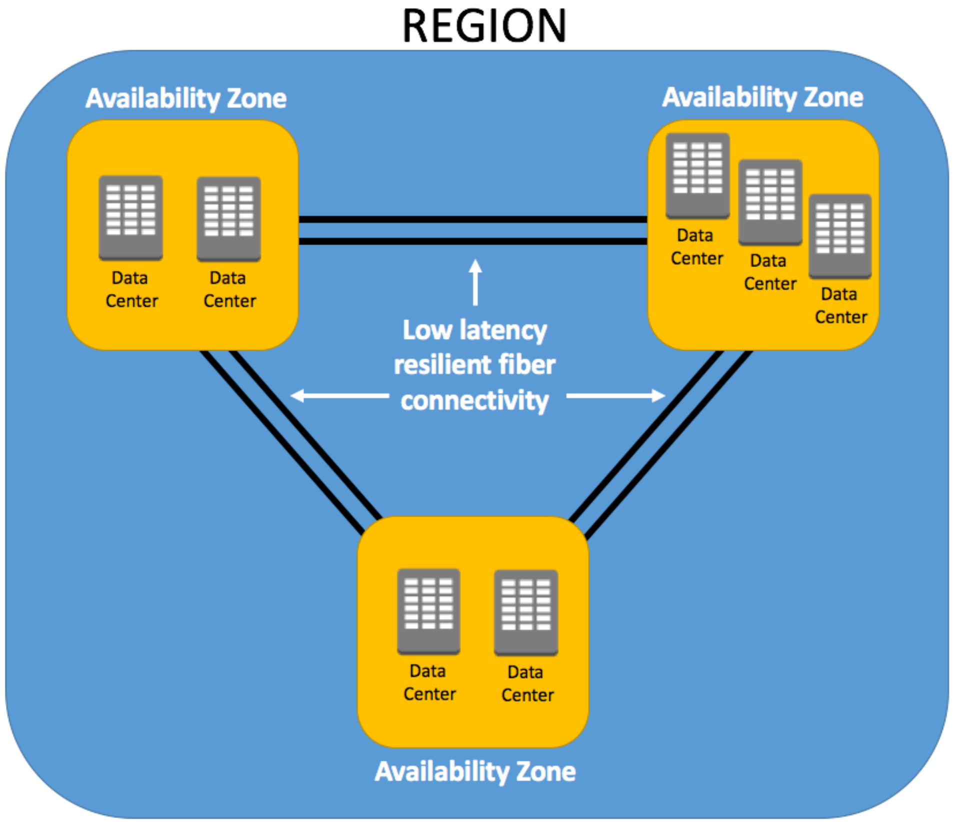
- Visit: https://docs.aws.amazon.com/index.html?nc2=h_ql_doc_do

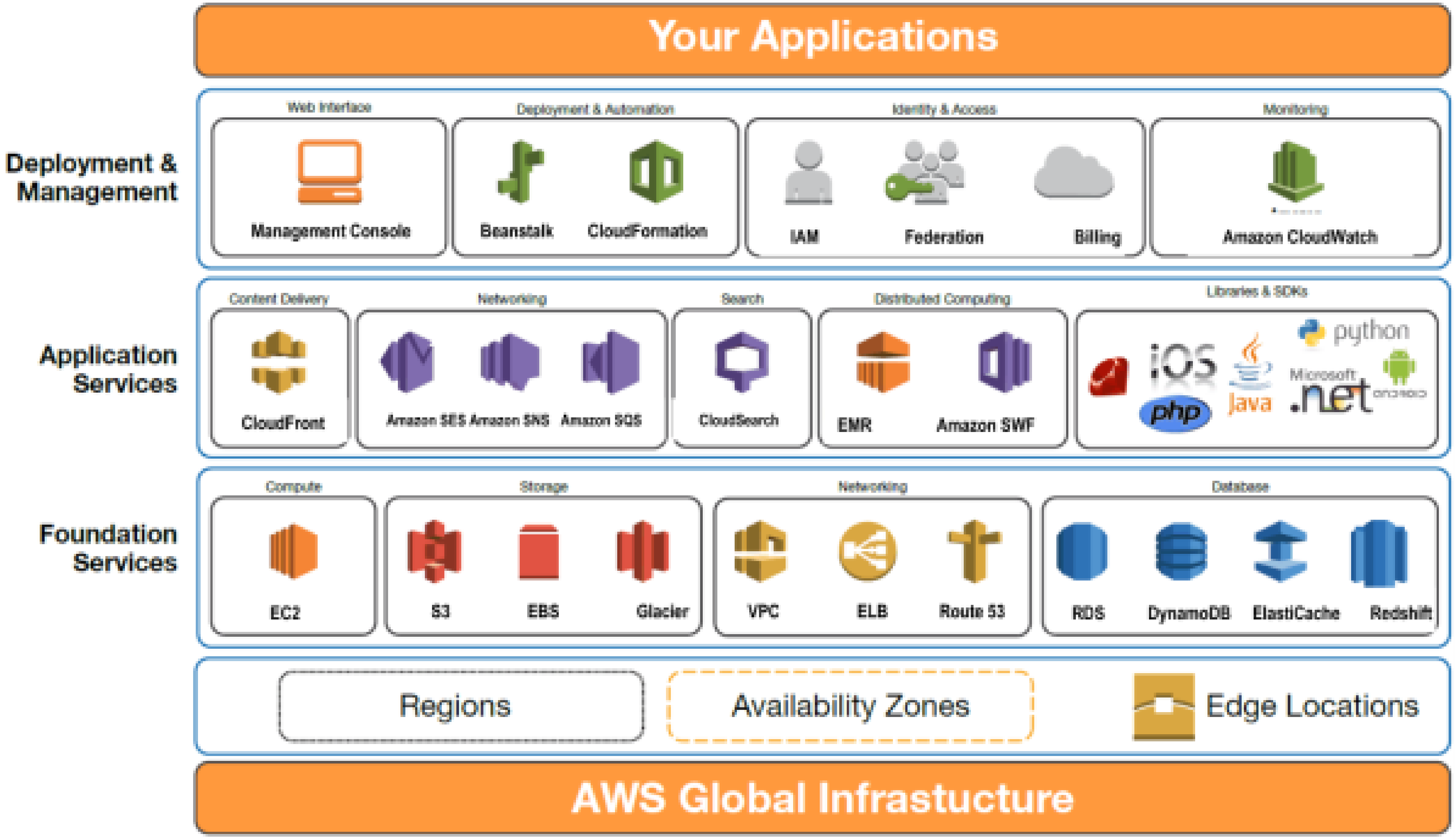
GLOBAL ARCHITECTURE

- 24 Launched Regions
- 3 Announced Regions
- 77 Availability Zones
- 2 Local Zones
- 5 Wavelength Zones
- 2x More Regions
- 245 Countries and Territories Served
- 97 Direct Connect Locations
- 216 Points of Presence
- 205 Edge Locations
- 11 Regional Edge Caches
- 130+ Services over 26 Subject area



- **Regions -** Independent Geographic Area
- **Available Zone -** Multiple isolated locations/Data Centers within a region
- An **edge location** is where end users access services located at AWS.





Your Applications

Deployment & Management

Web Interface
Management Console

Deployment & Automation
Beanstalk
CloudFormation

Identity & Access
IAM
Federation
Billing

Monitoring
Amazon CloudWatch

Application Services

Content Delivery
CloudFront

Networking
Amazon SES
Amazon SNS
Amazon SQS

Search
CloudSearch

Distributed Computing
EMR
Amazon SWF

Libraries & SDKs
iOS
php
Java
python
Microsoft .net
android

Foundation Services

Compute
EC2

Storage
S3
EBS
Glacier

Networking
VPC
ELB
Route 53

Database
RDS
DynamoDB
ElastiCache
Redshift

Regions
Availability Zones
Edge Locations

AWS Global Infrastucture

CLOUD SOFTWARE ENVIRONMENTS

EUCALYPTUS

- Eucalyptus is an open source software platform to implement Infrastructure as a Service (IaaS) in a private or hybrid cloud computing environment.
- The Eucalyptus cloud platform groups existing virtualized infrastructure to create resources in the cloud for infrastructure as a service, storage as a service, and the network as a service.
- The name Eucalyptus is an acronym for Elastic Utility Computing Architecture to link your programs to useful systems.

- At the University of California, Santa Barbara, Eucalyptus was founded from a research project in the Department of Computer Science and in 2009 became a profitable business called Eucalyptus Systems.
- In March 2012 Eucalyptus Systems declare a formal agreement with Amazon Web Services (AWS) to allow administrators to transfer instances between an Amazon Elastic Compute Cloud (EC2) and private Eucalyptus cloud to create a hybrid cloud Could.
- The partnership also allows Eucalyptus to work with Amazon product teams to develop unique features compatible with AWS.

EUCALYPTUS FEATURES INCLUDE

Supports virtual machines (VM) of Linux and Windows.

Application program interface (API) compatible with the Amazon EC2 platform.

Compatible with Simple Storage Service (S3) and Amazon Web Services (AWS).

It works with multiple hypervisors, including VMware, Xen and KVM.

- It can be installed and implemented from source code or DEB and RPM packages.
- Internal process communications are secured through SOAP and WS-Security.
- Multiple clusters can be virtualized as a single cloud.
- Administrative functions such as management and reports of users and groups.

VERSION 4.4.3 HAS THE FOLLOWING FEATURES

- Auto Scaling: allows application developers to scale Eucalyptus resources up or down according to the policies defined with the APIs and tools compatible with Amazon EC2
- Elastic Load Balancing: AWS-compatible service that provides greater fault tolerance for applications
- CloudWatch: an AWS-compatible service that allows users to collect metrics, set alarms, identify trends and take action to ensure applications run smoothly.
- Resource labeling: detailed reports for showback and chargeback scenarios; allows IT / DevOps to create reports that show the use of the cloud by application, department or user
- Extended instance types: expanded set of instance types to align more closely with those available in Amazon EC2. It was 5 before, now up to 15 types of instance.
- Maintenance mode: allows the replication of the hard disk of a virtual machine, the evacuation of the server node and provides a maintenance window.

Open
Nebula

- OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures.
- The OpenNebula platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of Infrastructure as a Service.
- The two primary uses of the OpenNebula platform:
  - Data center virtualization
  - Cloud deployments based on the KVM hypervisor, LXD system containers, and AWS Firecracker microVMs.

- The platform is also capable of offering the cloud infrastructure necessary to operate a cloud on top of existing VMware infrastructure.
- In early June 2020, OpenNebula announced the release of a new Enterprise Edition for corporate users, along with a Community Edition.
- OpenNebula CE is free and open-source software, released under the Apache License version 2.
- OpenNebula CE comes with free access to maintenance releases but with upgrades to new minor/major versions only available for users with non-commercial deployments or with significant contributions to the OpenNebula Community.
- OpenNebula EE is distributed under a closed-source license and requires a commercial Subscription.

BASIC COMPONENTS

- OpenNebula Internal Architecture
- Host: Physical machine running a supported hypervisor.
- Cluster: Pool of hosts that share datastores and virtual networks.
- Template: Virtual Machine definition.
- Image: Virtual Machine disk image.
- Virtual Machine: Instantiated Template. A Virtual Machine represents one life-cycle, and several Virtual Machines can be created from a single Template.
- Virtual Network: A group of IP leases that VMs can use to automatically obtain IP addresses. It allows the creation of Virtual Networks by mapping over the physical ones. They will be available to the VMs through the corresponding bridges on hosts.

BASIC COMPONENTS

- Virtual network can be defined in three different parts:
  - Underlying of physical network infrastructure.
  - The logical address space available (IPv4, IPv6, dual stack).
  - Context attributes (e.g. net mask, DNS, gateway). OpenNebula also comes with a Virtual Router appliance to provide networking services like DHCP, DNS etc.

PROJECT'S DEPLOYMENT MODEL

- The OpenNebula Project's deployment model resembles classic cluster architecture which utilizes
  - A front-end (master node)
  - Hypervisor enabled hosts (worker nodes)
  - Datastores
  - A physical network

FRONT-END MACHINE

- The master node, sometimes referred to as the front-end machine, executes all the OpenNebula services.
- This is the actual machine where OpenNebula is installed. OpenNebula services on the front-end machine include
  - the management daemon (oned),
  - scheduler (sched),
  - the web interface server (Sunstone server), and other advanced components.
- These services are responsible for queuing, scheduling, and submitting jobs to other machines in the cluster.
- The master node also provides the mechanisms to manage the entire system.
- Much of this is possible due to a monitoring subsystem which gathers information such as host status, performance, and capacity use.

HYPERVISOR ENABLED-HOSTS

- The worker nodes, or hypervisor enabled-hosts, provide the actual computing resources needed for processing all jobs submitted by the master node.
- OpenNebula hypervisor enabled-hosts use a virtualization hypervisor such as Vmware, Xen, or KVM.
- The KVM hypervisor is natively supported and used by default. Virtualization hosts are the physical machines that run the virtual machines and various platforms can be used with OpenNebula.
- A Virtualization Subsystem interacts with these hosts to take the actions needed by the master node.

STORAGE

- The datastores simply hold the base images of the Virtual Machines.
- The datastores must be accessible to the front-end; this can be accomplished by using one of a variety of available technologies or direct attached storage.
- Three different datastore classes are included with OpenNebula,
  - System datastores - Hold the images used for running the virtual machines. The images can be complete copies of an original image, deltas, or symbolic links depending on the storage technology used.
  - Image datastores - used to store the disk image repository. Images from the image datastores are moved to or from the system datastore when virtual machines are deployed or manipulated.
  - File datastores - Used for regular files and is often used for kernels, ram disks, or context files.

PHYSICAL NETWORKS

- Physical networks are required to support the interconnection of storage servers and virtual machines in remote locations.
- It is also essential that the front-end machine can connect to all the worker nodes or hosts.
- At the very least two physical networks are required as OpenNebula requires a service network and an instance network.
- The front-end machine uses the service network to access hosts, manage and monitor hypervisors, and to move image files.
- The instance network allows the virtual machines to connect across different hosts.
- The network subsystem of OpenNebula is easily customizable to allow easy adaptation to existing data centers.

openstack®

- OpenStack is a free open standard cloud computing platform, mostly deployed as infrastructure-as-a-service (IaaS) in both public and private clouds where virtual servers and other resources are made available to users.
- The software platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout a data center.
- Users either manage it through a web-based dashboard, through command-line tools, or through RESTful web services.
- OpenStack began in 2010 as a joint project of Rackspace Hosting and NASA. As of 2012, it was managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012 to promote OpenStack software and its community.

COMPONENTS

- Compute (Nova):
  - Nova is the OpenStack project that provides a way to provision compute instances (aka virtual servers). Nova supports creating virtual machines, baremetal servers (through the use of ironic), and has limited support for system containers. Nova runs as a set of daemons on top of existing Linux servers to provide that service.

- Networking (Neutron)
  - Neutron is an OpenStack project to provide "network connectivity as a service" between interface devices (e.g., vNICs) managed by other OpenStack services (e.g., nova). It implements the OpenStack Networking API.

- Block storage (Cinder)
- Cinder is the OpenStack Block Storage service for providing volumes to Nova virtual machines, Some of the goals of Cinder are to be/have:
  - Component based architecture: Quickly add new behaviors
  - Highly available: Scale to very serious workloads
  - Fault-Tolerant: Isolated processes avoid cascading failures
  - Recoverable: Failures should be easy to diagnose, debug, and rectify
  - Open Standards: Be a reference implementation for a community-driven api
- Identity (Keystone)
  - Keystone is an OpenStack service that provides API client authentication, service discovery, and distributed multi-tenant authorization by implementing OpenStack's Identity API. It is the common authentication system across the cloud operating system.

- Image (Glance)
  - The Image service (glance) project provides a service where users can upload and discover data assets that are meant to be used with other services. This currently includes images and metadata definitions.
    - Images - Glance image services include discovering, registering, and retrieving virtual machine (VM) images.
    - Metadata Definitions - Glance hosts a metadefs catalog. This provides the OpenStack community with a way to programmatically determine various metadata key names and valid values that can be applied to OpenStack resources.
- Object storage (Swift)
  - Swift is a distributed, eventually consistent object/blob store. The OpenStack Object Store project, known as Swift, offers cloud storage software so that you can store and retrieve lots of data with a simple API. It's built for scale and optimized for durability, availability, and concurrency across the entire data set.

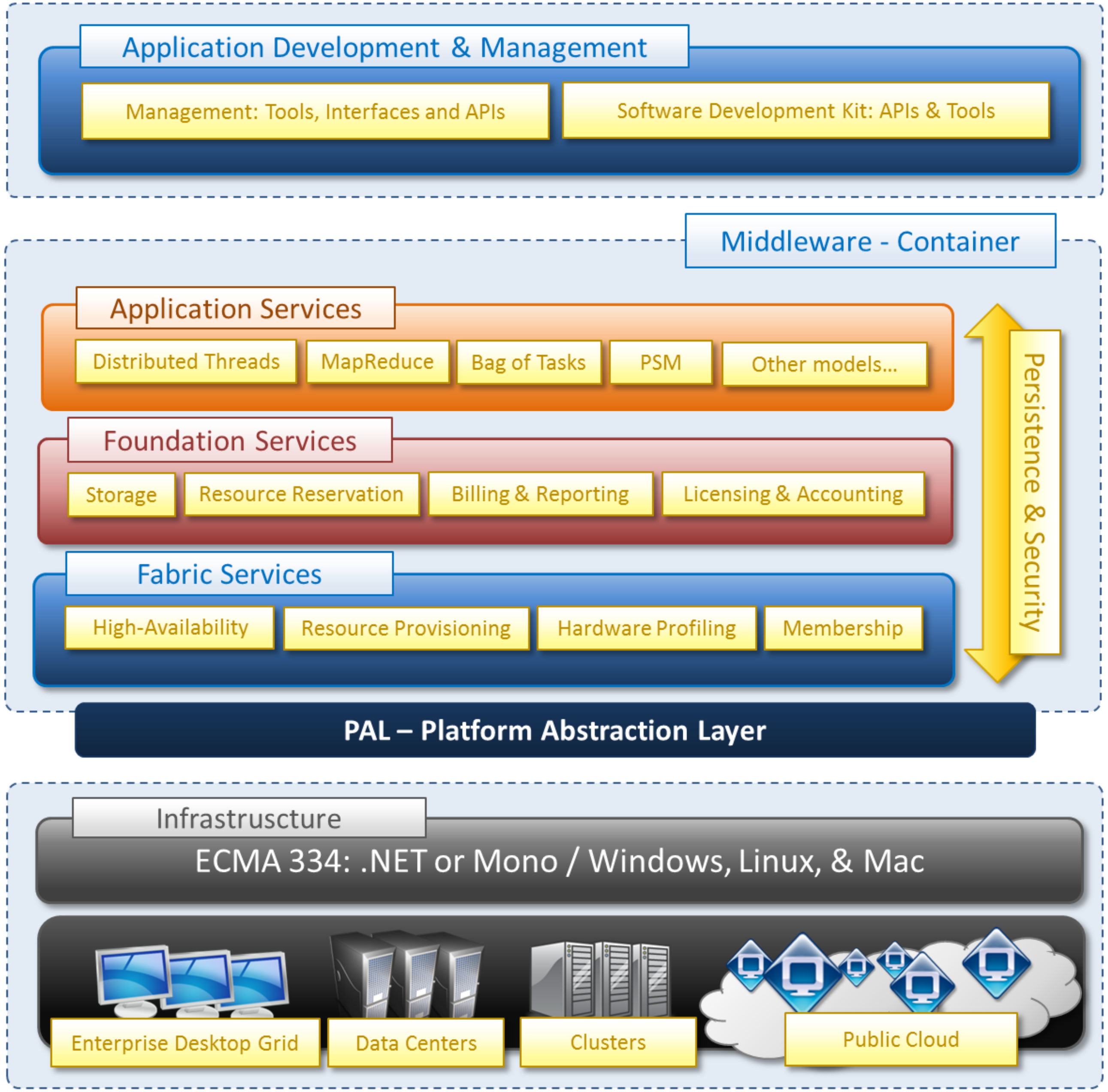
- Dashboard (Horizon)
  - Horizon is the canonical implementation of OpenStack's Dashboard, which provides a web based user interface to OpenStack services including Nova, Swift, Keystone, etc.
- Orchestration (Heat)
  - Heat is a service to orchestrate multiple composite cloud applications using templates, through both an OpenStack-native REST API and a CloudFormation-compatible Query API
- Workflow (Mistral)
  - Mistral is a service that manages workflows. User typically writes a workflow using workflow language based on YAML and uploads the workflow definition to Mistral via its REST API. Then user can start this workflow manually via the same API or configure a trigger to start the workflow on some event

OTHER COMPONENTS

- Telemetry (Ceilometer)
- Database (Trove)
- Elastic map reduce (Sahara)
- Bare metal (Ironic)
- Messaging (Zaqar)
- Shared file system (Manila)
- DNS (Designate)
- Search (Searchlight)
- Key manager (Barbican)
- Container orchestration (Magnum)
- Root Cause Analysis (Vitrage)
- Rule-based alarm actions (Aodh)

ANEKA

- Aneka is a platform and a framework for developing distributed applications on the Cloud.
- Aneka provides developers with a rich set of APIs for transparently exploiting such resources and expressing the business logic of applications by using the preferred programming abstractions.
- System administrators can leverage on a collection of tools to monitor and control the deployed infrastructure.
- This can be a public cloud available to anyone through the Internet, or a private cloud constituted by a set of nodes with restricted access.



Application Development & Management

Management: Tools, Interfaces and APIs

Software Development Kit: APIs & Tools

Middleware - Container

Application Services

Distributed Threads

MapReduce

Bag of Tasks

PSM

Other models...

Foundation Services

Storage

Resource Reservation

Billing & Reporting

Licensing & Accounting

Fabric Services

High-Availability

Resource Provisioning

Hardware Profiling

Membership

Persistence & Security

PAL – Platform Abstraction Layer

Infrastrucsture

ECMA 334: .NET or Mono / Windows, Linux, & Mac

Enterprise Desktop Grid

Data Centers

Clusters

Public Cloud

CLOUDSIM

- CloudSim is a simulation toolkit that supports the modeling and simulation of the core functionality of cloud, like job/task queue, processing of events, creation of cloud entities(datacenter, datacenter brokers, etc), communication between different entities, implementation of broker policies, etc.
- This toolkit allows to:
  - Test application services in a repeatable and controllable environment.
  - Tune the system bottlenecks before deploying apps in an actual cloud.
  - Experiment with different workload mix and resource performance scenarios on simulated infrastructure for developing and testing adaptive application provisioning techniques

- Core features of CloudSim are:
  - The Support of modeling and simulation of large scale computing environment as federated cloud data centers, virtualized server hosts, with customizable policies for provisioning host resources to virtual machines and energy-aware computational resources
  - It is a self-contained platform for modeling cloud's service brokers, provisioning, and allocation policies.
  - It supports the simulation of network connections among simulated system elements.
  - Support for simulation of federated cloud environment, that inter-networks resources from both private and public domains.
  - Availability of a virtualization engine that aids in the creation and management of multiple independent and co-hosted virtual services on a data center node.
  - Flexibility to switch between space shared and time shared allocation of processing cores to virtualized services.
- For more information: https://www.cloudsimtutorials.online/cloudsim/#:~:text=CloudSim%20is%20a%20simulation%20toolkit,implementation%20of%20broker%20policies%2C%20etc.