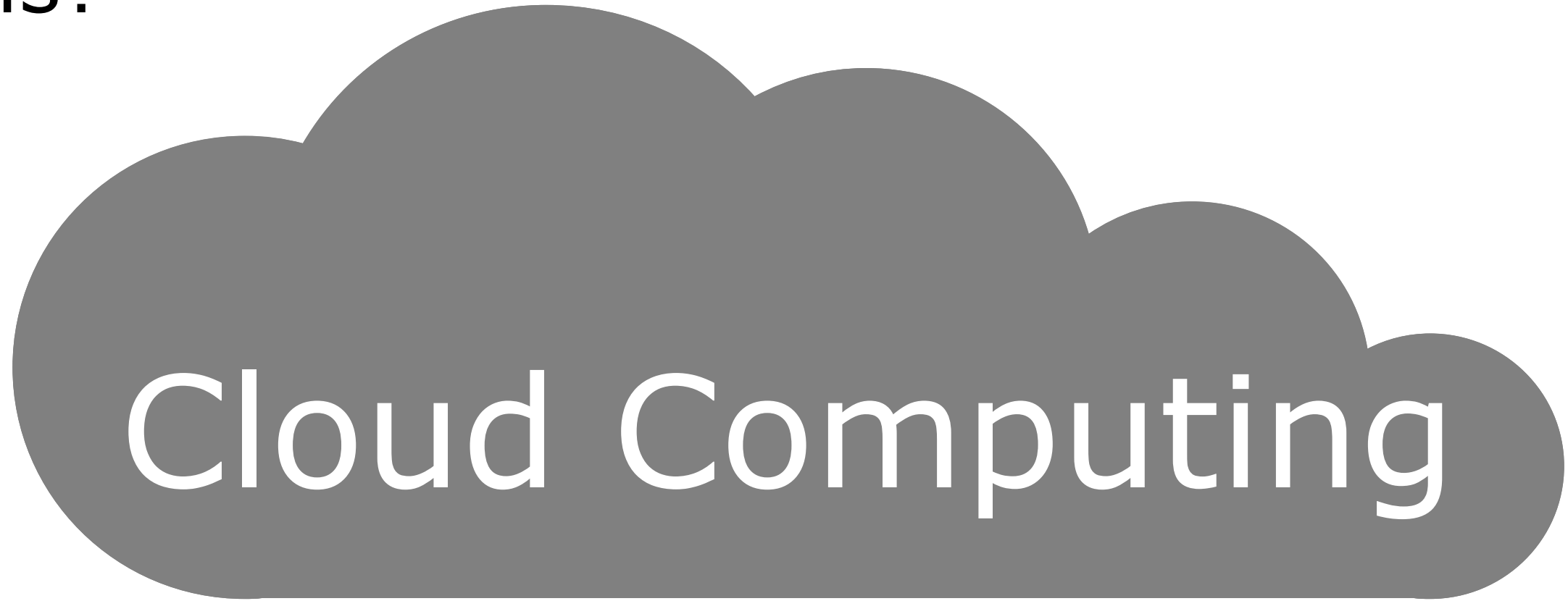
Cloud Computing

Cloud Computing Overview

What do you think Cloud Computing is?



What is Cloud Computing?

"The practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer."

Oxford Dictionary

What is Cloud Computing?

"The practice of storing regularly used computer data on multiple servers that can be accessed through the Internet."

Webster Dictionary

MACHINE INTELLIGENCE 3.0

ENTERPRISE INTELLIGENCE

VISUAL
Orbital Insight, planet, clarifai, DEEPVISION, cortica, Iqocion, SPACE_KNOW, Captricity, netra, deepomatic

AUDIO
Gridspace, TalkIQ, nexidia, twilio, CAPIO, Expect Labs, Clover, Mobvoi, Qurious.AI, popup archive

SENSOR
PREDIX, C3IoT, MAANA, Sentenai, PLANET OS, UPTAKE, IMUBIT, Preferred Networks, thingworx, KONUX, Alluvium

INTERNAL DATA
PRIMER, IBM WATSON, Cycorp, Palantir, ARIMO, Alation, sapho, Outlier, Digital Reasoning

MARKET
mattermark, Quid, DataFox, PREMISE, Bottlenose, MOTIVA, enigma, CBINSIGHTS, Tracxn, predata

ENTERPRISE FUNCTIONS

CUSTOMER SUPPORT
DigitalGenius, Kasisto, ELOQUENT, Wiseio, ACTIONIQ, zendesk, Preact, CLARABRIDGE

SALES
collective[i], 6sense, fusemachines, AVISO, salesforce, INSIDE SALES.COM, clari, Zensight

MARKETING
MINTIGO, Lattice, RADIUS, LiftIgniter, [PERSADO], brightfunnel, retention, MOTIVA, COGNICOR, AIRPR, msg.ai

SECURITY
CYLANCE, DARKTRACE, ZIMPERIUM, deepinstinct, Sentinel, DEMISTO, graphistry, drawbridge, SignalSense, AppZen

RECRUITING
textio, entelo, Wade & Wendy, hi, unitive, SpringRole, GIGSTER, HireVue

AUTONOMOUS SYSTEMS

GROUND NAVIGATION
drive.ai, AdasWorks, ZOOX, MOBILEYE, UBER, Google, TESLA, nuTonomy, Auro Robotics

AERIAL
SKYDIO, SHIELD AI, Airware, dji, LILY, DroneDeploy, pilot.ai, SKYCATCH

INDUSTRIAL
JAYBRIDGE, OSARO, CLEARPATH, fetch, KINDRED, HARVEST, rethink robotics

AGENTS

PERSONAL
amazon alexa, Cortana, Allo, facebook, Siri, Replika

PROFESSIONAL
butter.ai, pogo, SKIPFLAG, clara, x.ai, slack, talla, Zoom.ai, sudo

INDUSTRIES

AGRICULTURE
BLUE RIVER, mavrx, tule, TRACE GENOMICS, Pivot Bio, TerrAvion, AGRI-DATA, Descartes Labs, udio, abundant ROBOTICS

EDUCATION
KNEWTON, volley, gradescope, CTI, coursera, UDACITY, altschool

INVESTMENT
Bloomberg, sentient, iSENTIUM, KENSHO, alphasense, Dataminr, CEREBELLUM CAPITAL, Quandl

LEGAL
blueJ, BEAGLE, Everlaw, RAVEL, seal, ROSS, LEGAL ROBOT

LOGISTICS
NAUTO, Acerta, PRETECKT, clearmetal, Routific, MARBLE, PITSTOP

INDUSTRIES CONT'D

MATERIALS
zymergen, Citrine, Eigen Innovations, SIGHT MACHINE, GINKGO BIOWORKS, nanotronics, CALCULARIO

RETAIL FINANCE
TALA, zestfinance, Lendo, earnest, affirm, MIRADOR, wealthfront, Betterment

HEALTHCARE

PATIENT
PULSE, CareSkore, ZEPHYR HEALTH, IBM Watson Health, OncoRx, SENTRIAN, Atomwise, Numerate

IMAGE
BUTTERFLY, 3SCAN, ARTERYS, enlitic, BAYLABS, imagia, Google DeepMind

BIOLOGICAL
iCarbonX, color, GRAIL, deep genomics, RECURSION, LUMINIST, Numerate, Atomwise, verily, WHOLE BIOME

shivonzilis.com/MACHINEINTELLIGENCE · Bloomberg BETA

TECHNOLOGY STACK

AGENT ENABLERS
OCTANE.AI, howdy., Maluuba, KITT·AI, OpenAI Gym, Kasisto, AUTOMAT, semanticmachines

DATA SCIENCE
DOMINO, SPARKBEYOND, rapidminer, kaggle, DataRobot, ŷhat, AYASDI, data iku, seldon, yseop, bigml

MACHINE LEARNING
CognitiveScale, GoogleML, context relevant, Cycorp, HyperScience, narrologics, minds.ai, H_2O.ai, SCALED INFERENCE, sparkcognition, Loop, GEOMETRIC INTELLIGENCE, deepsense.io, reactive, skymind, bonsai

NATURAL LANGUAGE
agolo, AYLIEN, LEXALYTICS, Narrative Science, loop AI Labs, spaCy, LUMINOSO, cortical.io, MonkeyLearn

DEVELOPMENT
SIGOPT, HyperOpt, fuzzy.io, pkite, rainforest, lobe, Anodot, Signifai, LAYER 6 AI, bonsai

DATA CAPTURE
CrowdFlower, diffbot, CrowdAI, import.io, Paxata, DATASIFT, amazon mechanical turk, enigma, WorkFusion, DATALOGUE, TRIFACTA, parsehub

OPEN SOURCE LIBRARIES
Keras, Chainer, CNTK, TensorFlow, Caffe, H2O, DEEPLEARNING4J, theano, torch, DSSTNE, Scikit-learn, AzureML, neon, MXNet, DMTK, Spark, PaddlePaddle, WEKA

HARDWARE
KNUPATH, TENSTORRENT, Cirrascale, NVIDIA, intel nervana, Movidius, tensilica, GoogleTPU, 10^{26} Labs, Qualcomm, Cerebras, Isosemi

RESEARCH
OpenAI, maisense, ELEMENT AI, vicarious, KNOGGIN, Numenta, Kimera Systems, Cogitai

Cloud Computing Perspectives

Perspectives highly influenced by roles and responsibilities within an organization

- End-User
- Application Developer
- IT Infrastructure Manager
- CIO
- CFO
- Service Provider

What is Cloud Computing? – Take 2

Further perspectives include:

- “An approach to computing that’s about Internet scale and connecting to a variety of devices and endpoints.”
- “Treating hardware and software resources as a utility.”
- “A way to save a ton of money by only paying for what you need.”
- “A way to scale huge when you need something done fast.”

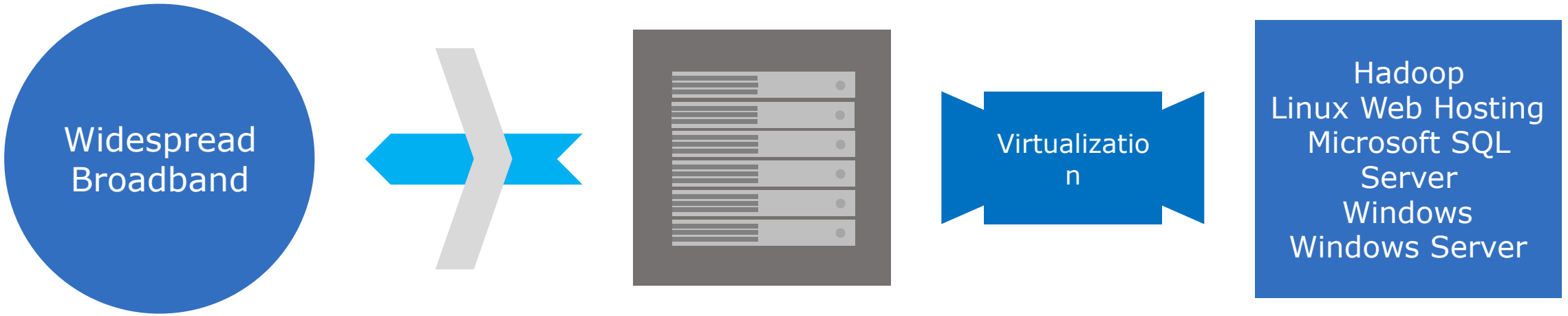
Evolution of Cloud Computing

Order of Evolution

Stage	Characteristics
Grid Computing	Solving large problems with parallel computing Made mainstream by Global Alliance
Utility Computing	Computing resources offered as a metered service Late 1990s
Software as a Service	Subscription-based software accessed over the Internet Gained momentum after 2001
Cloud Computing	Next-generation datacenters with virtualization technology Full stack of service - IasS, PaaS, & SaaS

Key Enabling Technologies

- Ubiquitous fast wide-area networks
- Powerful and inexpensive servers
- High-performance virtualization technology



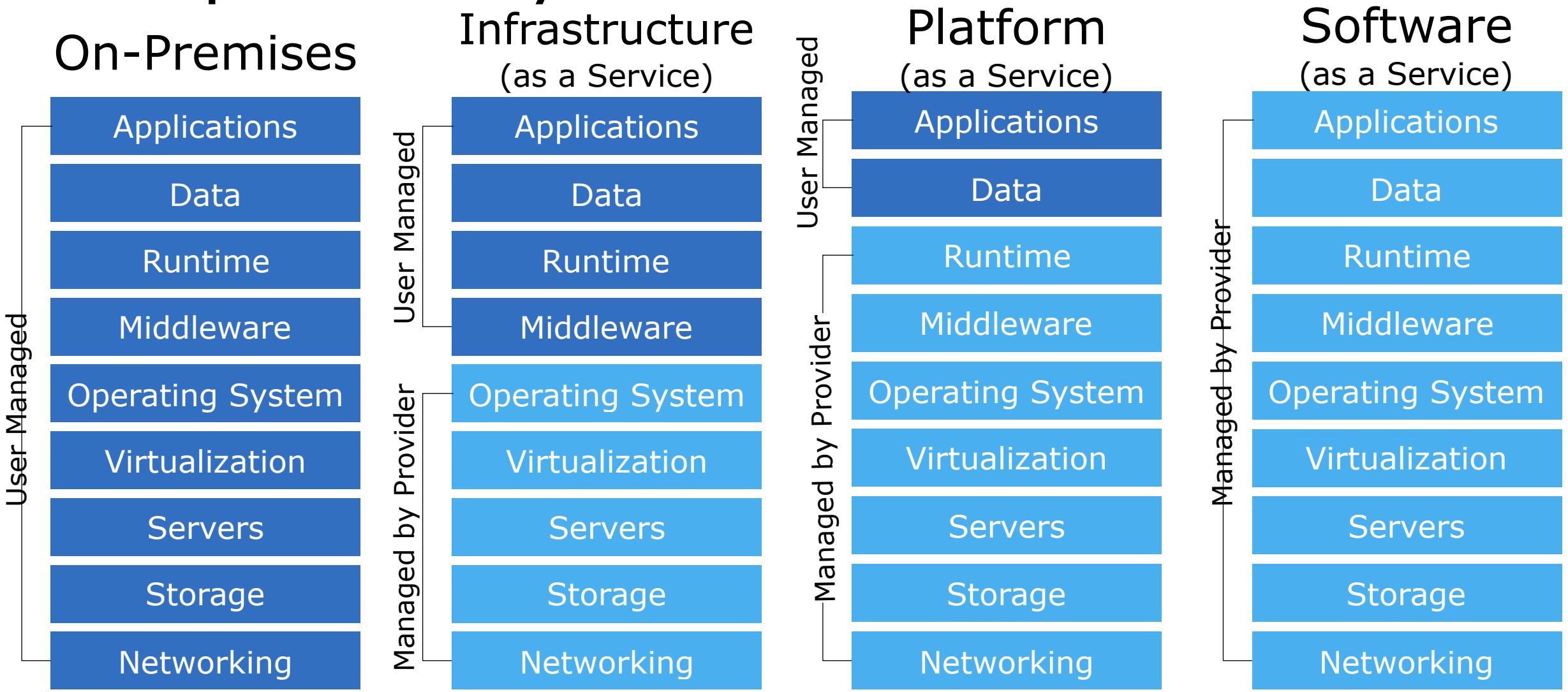
Five Key Cloud Characteristics

- On-demand self-service
- Ubiquitous network access
- Location-independent resource pooling
- Rapid elasticity
- Pay for what you use

Cloud Computing Service Models

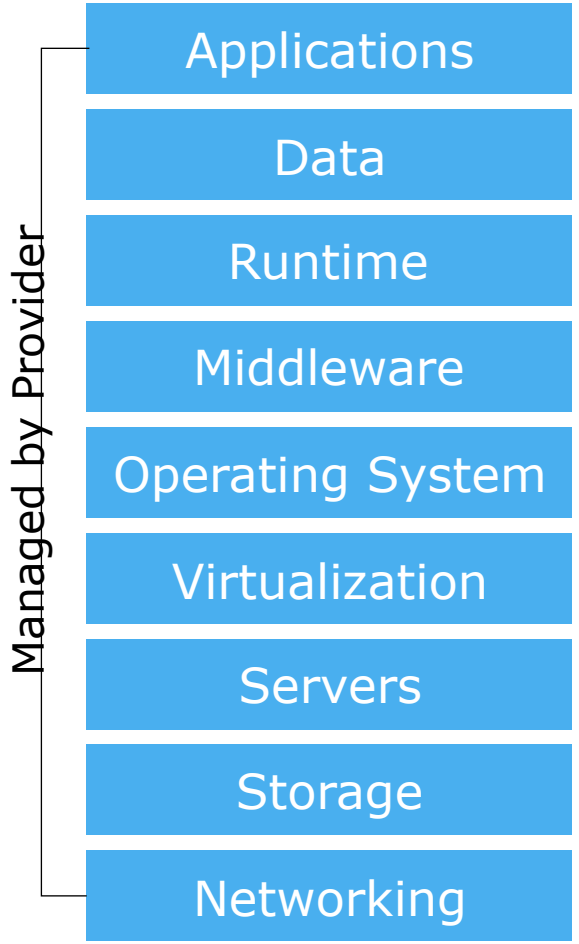
Model	Description
Software as a Service (SaaS)	Consume it End-User Applications delivered as a service, rather than by on-premises software
Platform as a Service (PaaS)	Build on it Application platform or middleware provided as a service on which developers can build and deploy custom applications
Infrastructure as a Service (IaaS)	Migrate to it Computing, storage, or other IT infrastructure provided as a service, rather than as a dedicated capability

Service Model Division of Responsibility



Software as a Service (SaaS)

- Internet hosted software
- Full vendor maintenance
- No upfront cost
- Pay for services as they are consumed



Salesforce.com as an SaaS

Online salesforce automation and CRM

The following are sales points advertised by Salesforce:

- No vendor lock-in
- No large up-front investment
- No maintenance headaches
- No steep learning curve
- No outdated solutions

Microsoft Office 365

Microsoft Office Suite on the Web

Flexible environment to collaborate and work together

- Multiple environments
  - Desktop, tablet and mobile phones
- Work anywhere, anytime, any device
- Concurrent edits

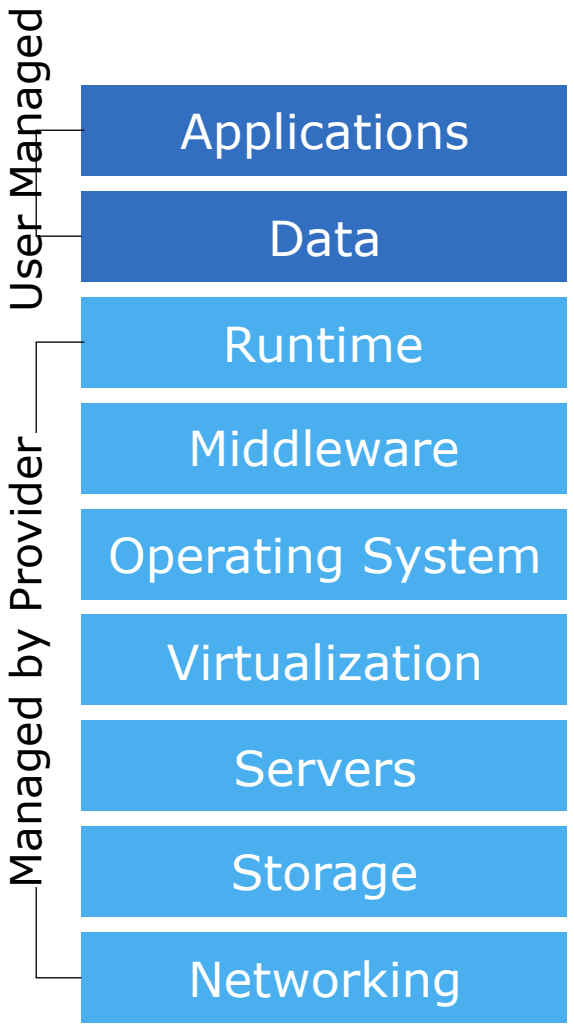
GoToMeeting, WebEx, Skype

Most SaaSs have their roots on the desktop

- Modern versions of old utilities
  - Think old TelCos
  - Services charged like utilities

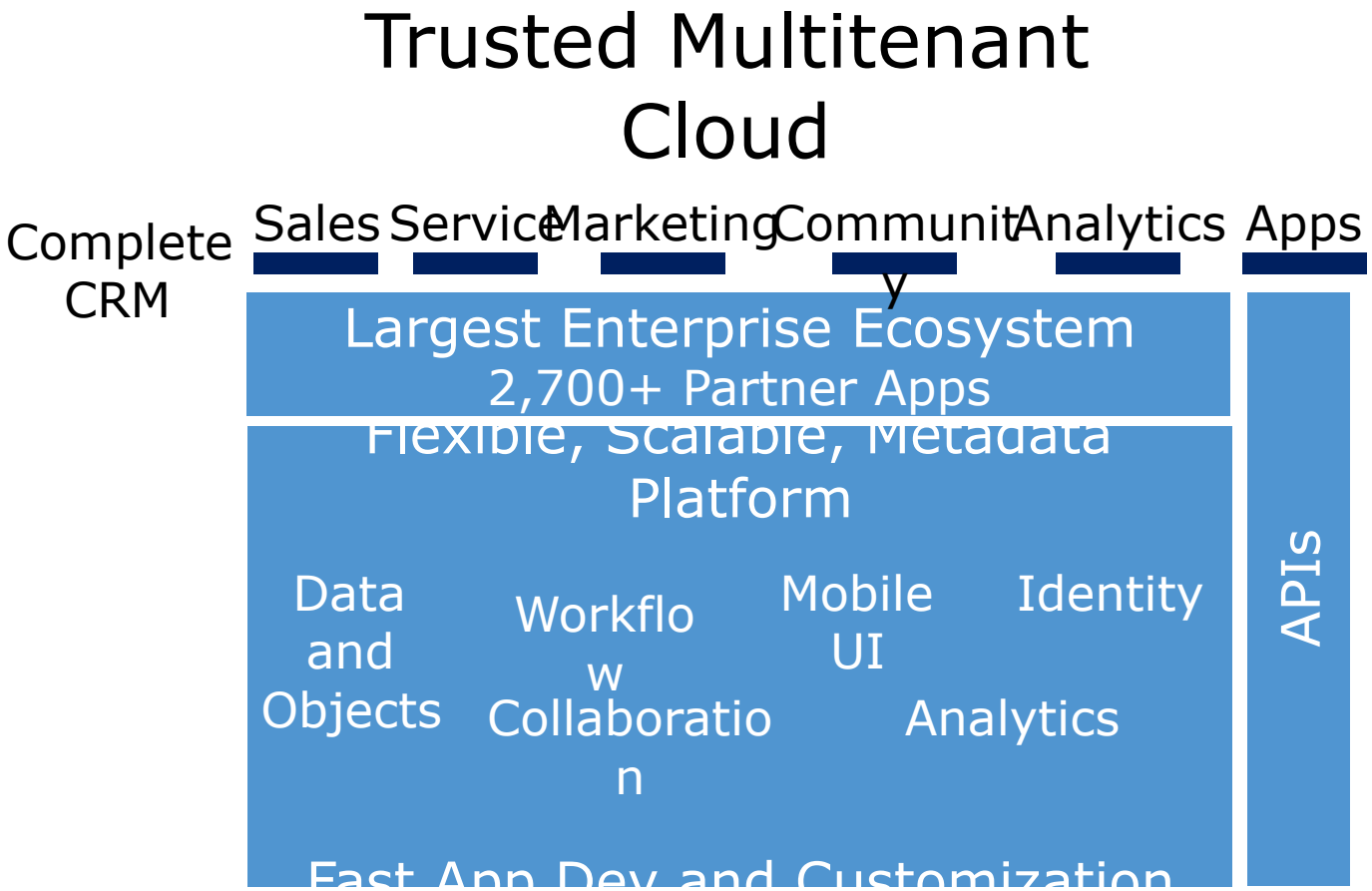
Platform as a Service (PaaS)

- Delivers and manages various development environments
- Environment and tools can be easily provisioned and torn down



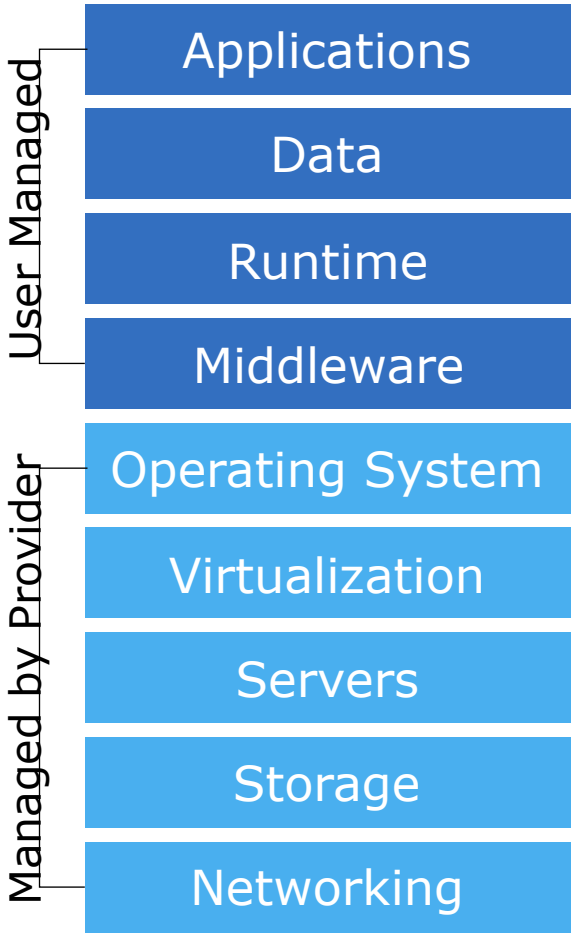
Salesforce.com as a PaaS (?)

- Multi-tenant cloud
  - Multiple user have shared resources as well as dedicated resources
- Metadata-driven architecture
  - All customizations (code, configuration, apps) are specified and saved as metadata
- API-to-CRM engine



Infrastructure as a Service (IaaS)

- Dedicated virtual machines (VMs)
- Users configure server type, operating system, storage, network, etc.
- Scale up and down



Storage – SaaS or IaaS?

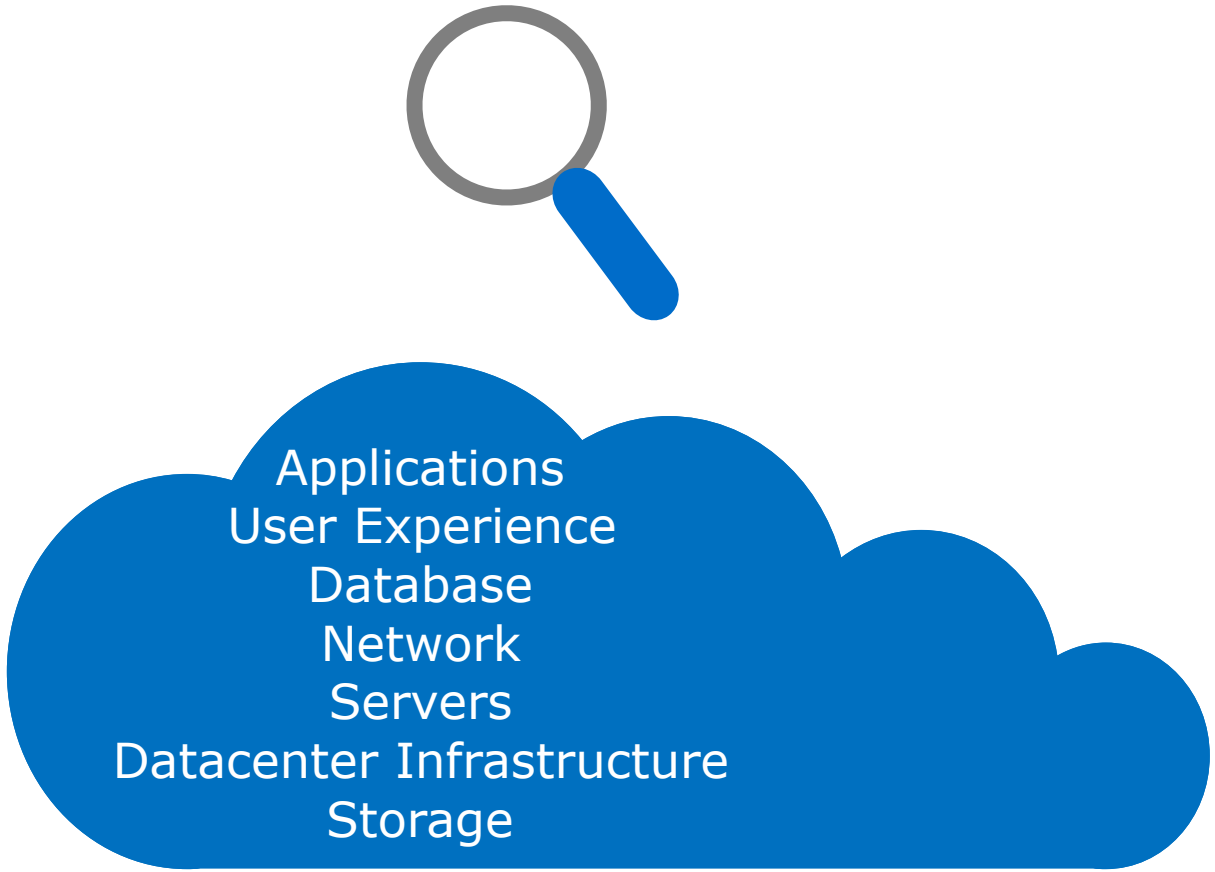
- Backup storage services such as Google Drive, Box, and OneDrive should be considered SaaS; user is not responsible for the backup software
- Storage services such as Azure Blobs and Amazon S3 are considered IaaS

Desktop (as a service) Cloud Service

- Provides and manages a virtual desktops
- Allows smaller companies who find Virtual Desktop Infrastructure (VDI) to be cost prohibitive to deliver similar services
- Quickly deploy new solutions across the entire enterprise located in multiple regions

Monitoring (as a service) Cloud Service

- Consists of tools and applications meant to monitor certain aspects of an application, server, system, or any other IT component
- State monitoring is the most common service
  - The state of a component is constantly evaluated and results are displayed in real time

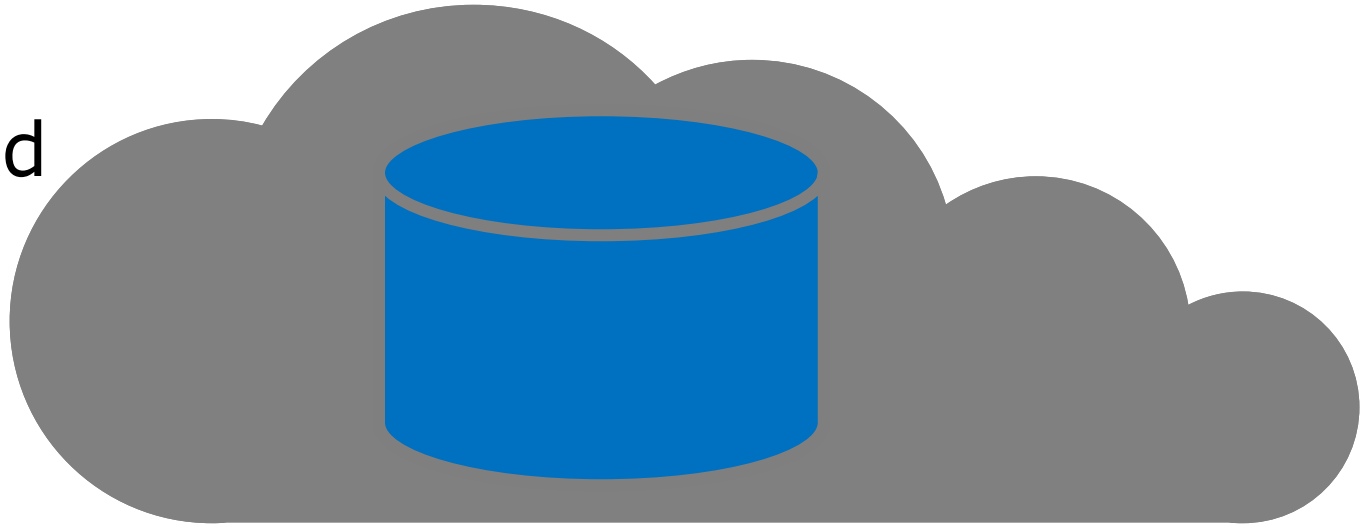


Communication (as a service) Cloud Service

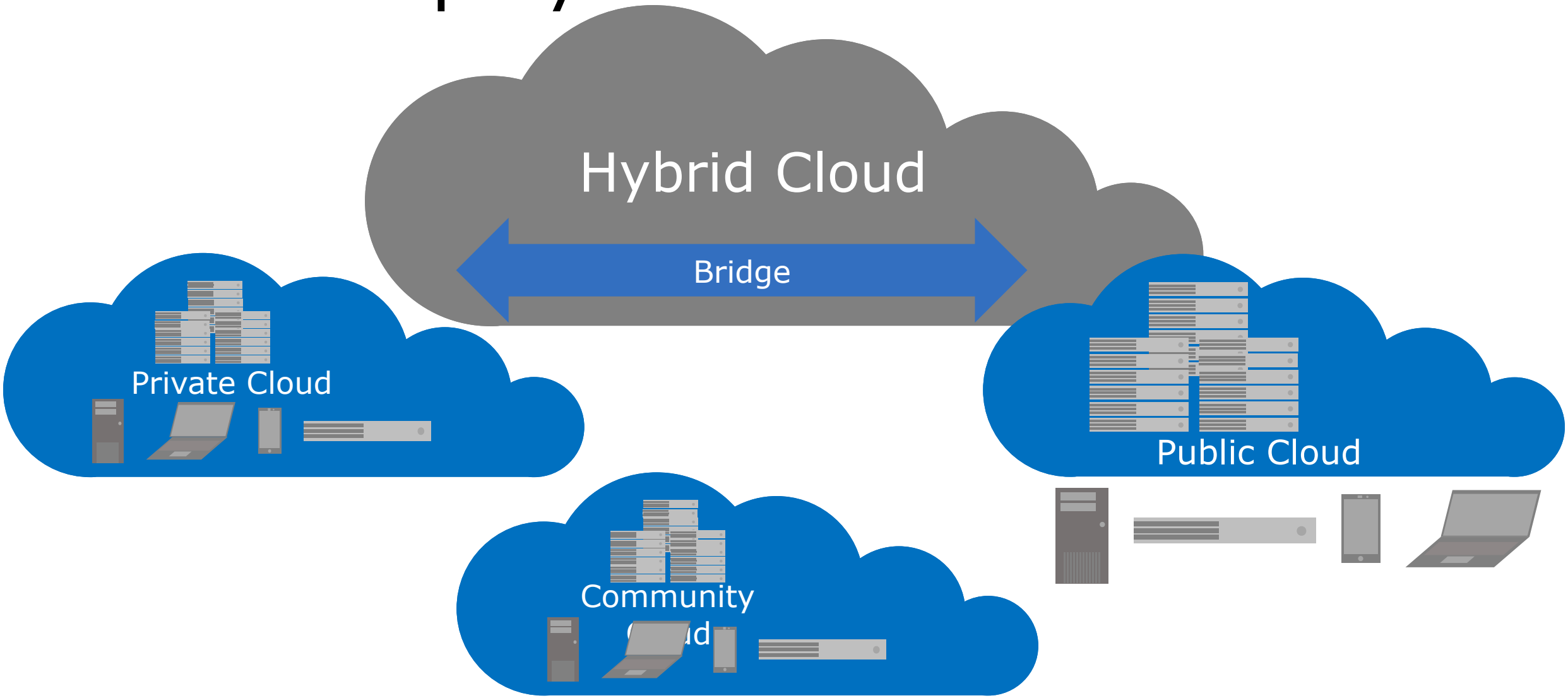
- Includes enterprise communication solutions such as VoIP (Voice over IP), instant messaging (IM) and video conferencing tha can be leased
- Vendor is responsible for all hardware and software and offers guaranteed Quality of Service

Database (as a service) Cloud Service

- Cloud-based approach to the storage and management of structured and unstructured data
- Rather than offering raw storage platforms, this service offers functionality of database platforms such as SQL Server, MySQL, Oracle, and NoSQL



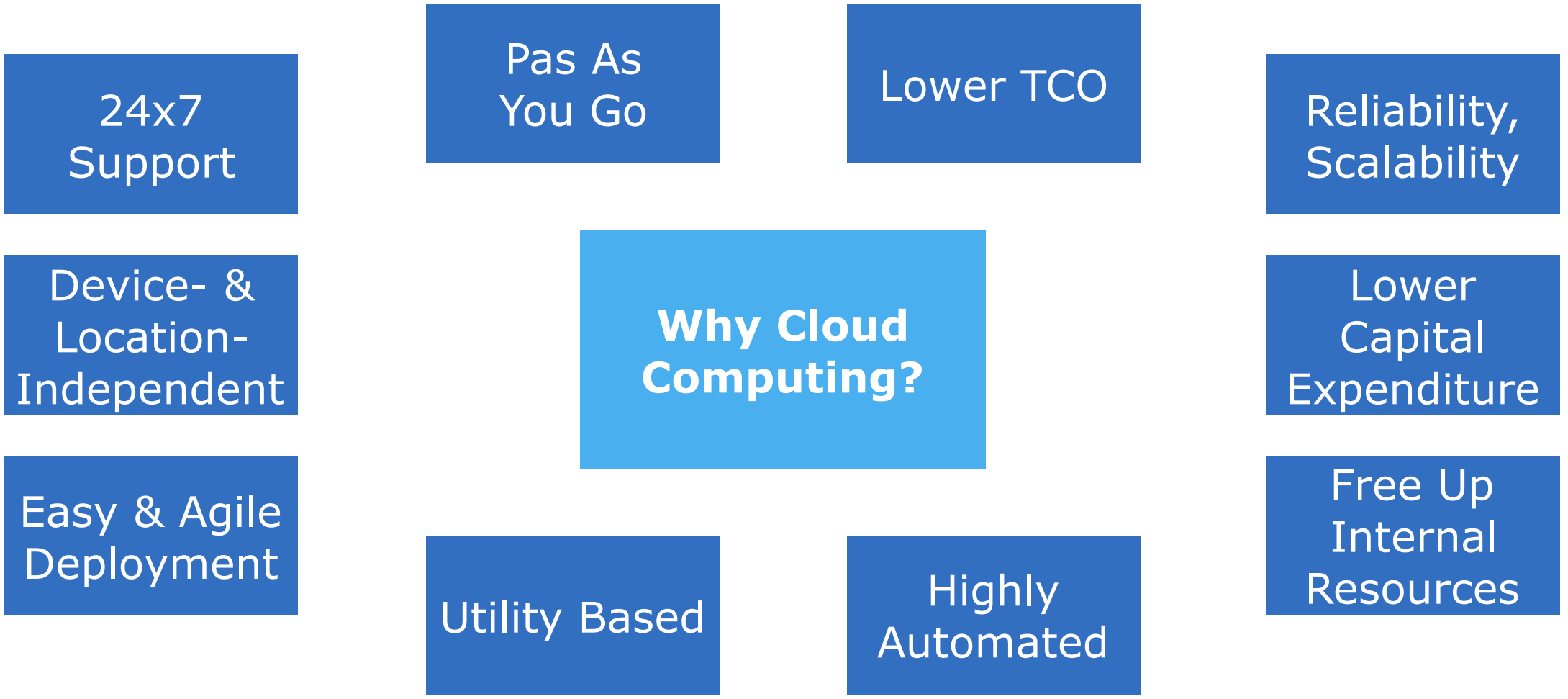
Cloud Deployment Model



Cloud Deployment Models – Advantages & Characteristics

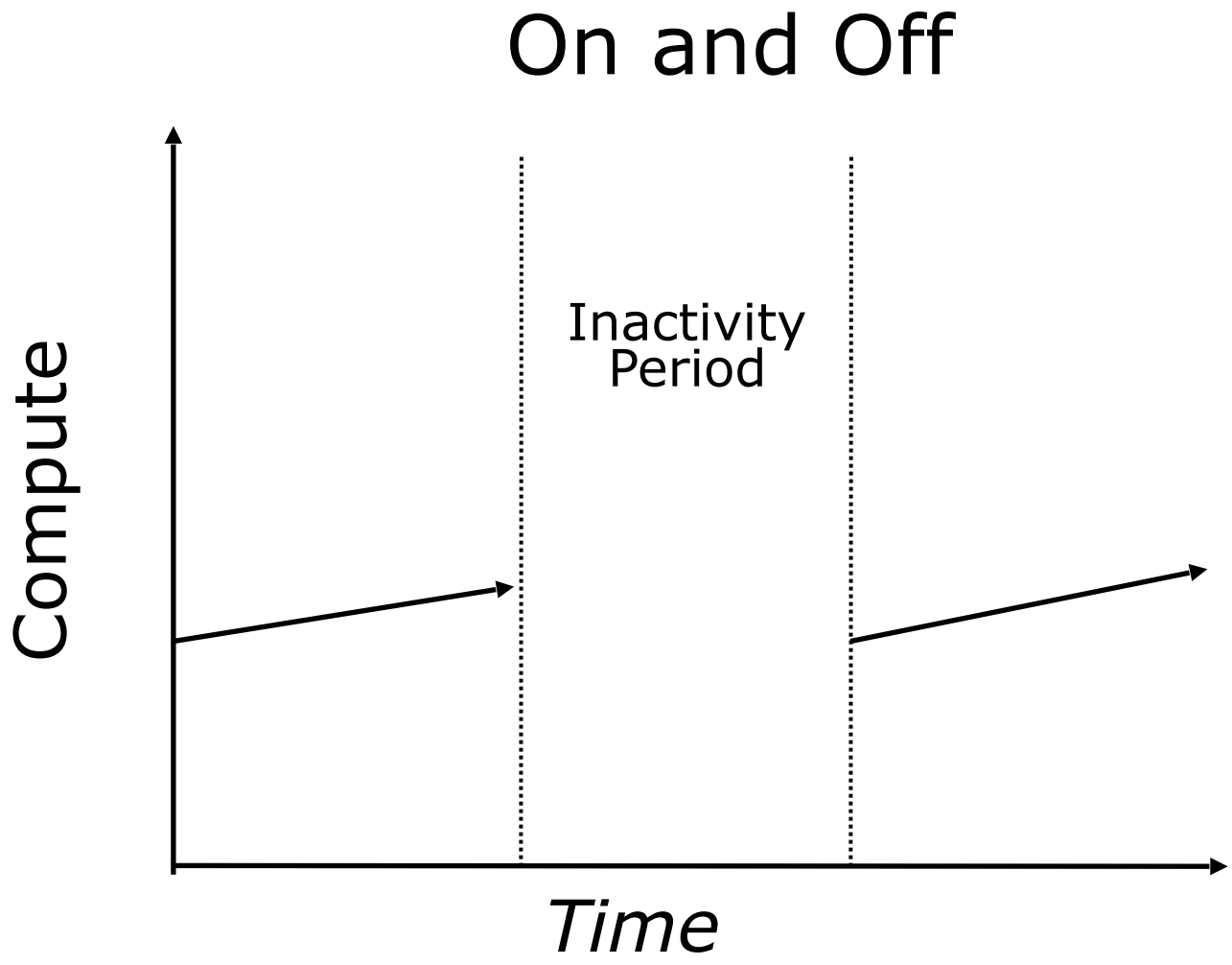
Model	Advantages and Characteristics
Public	Shifts capital expense to operating expense Offers pay-as-you-go pricing Supports multiple tenants
Private	Leverages existing capital expense Can help reduce operating costs Intended for a single tenant
Hybrid	Bridges one or more community, private, or public clouds Allows manipulation of CapEx and OpEx to optimize cost Supports resource portability
Community	Allows sharing of CapEx and OpEx to reduce costs Brings together groups with a common interest Supports resource portability

Why Cloud Computing?



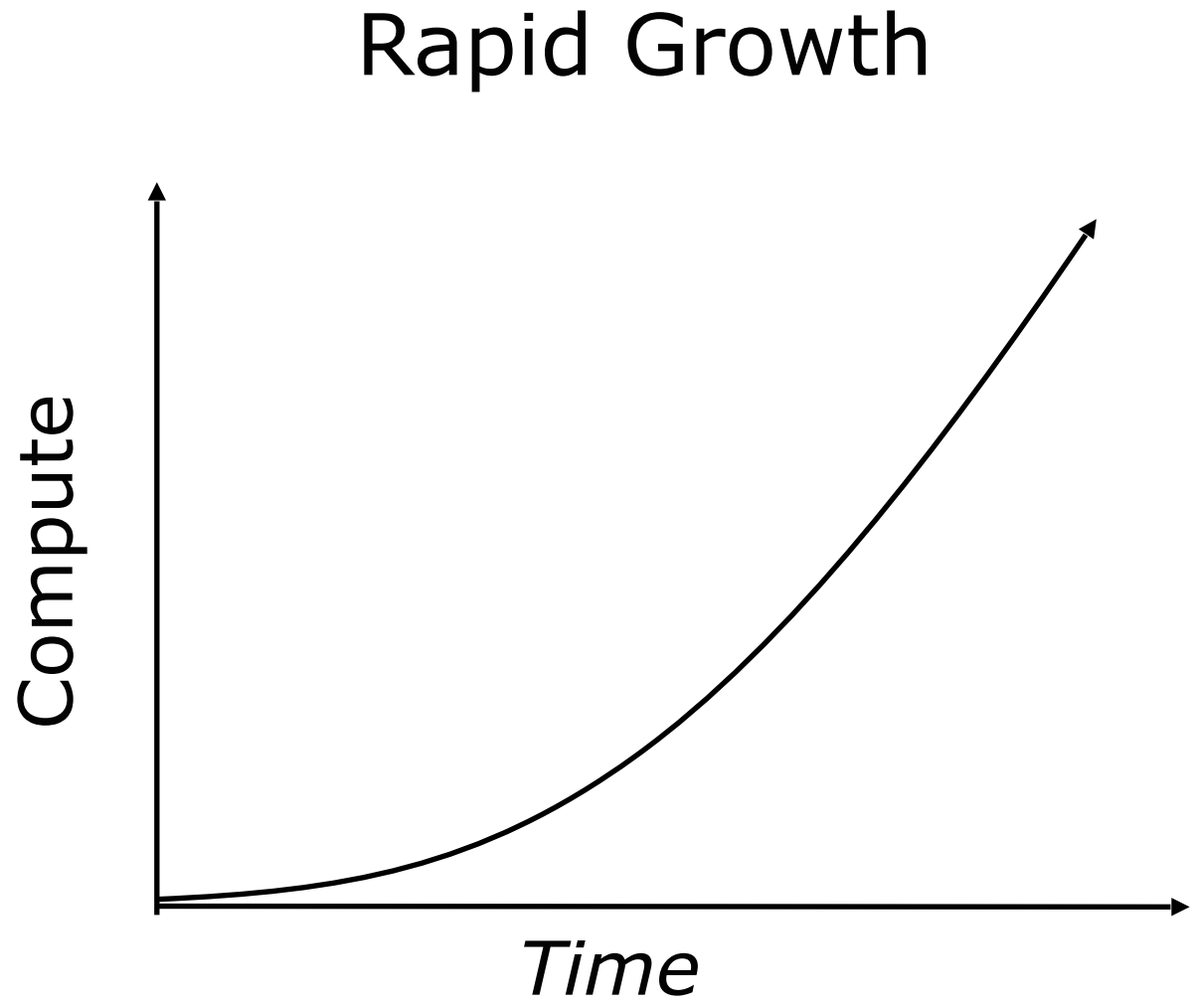
Typical Computing Pattern

- On & off workloads
  - Batch jobs
- Wasted Capacity
- Time to market can be cumbersome



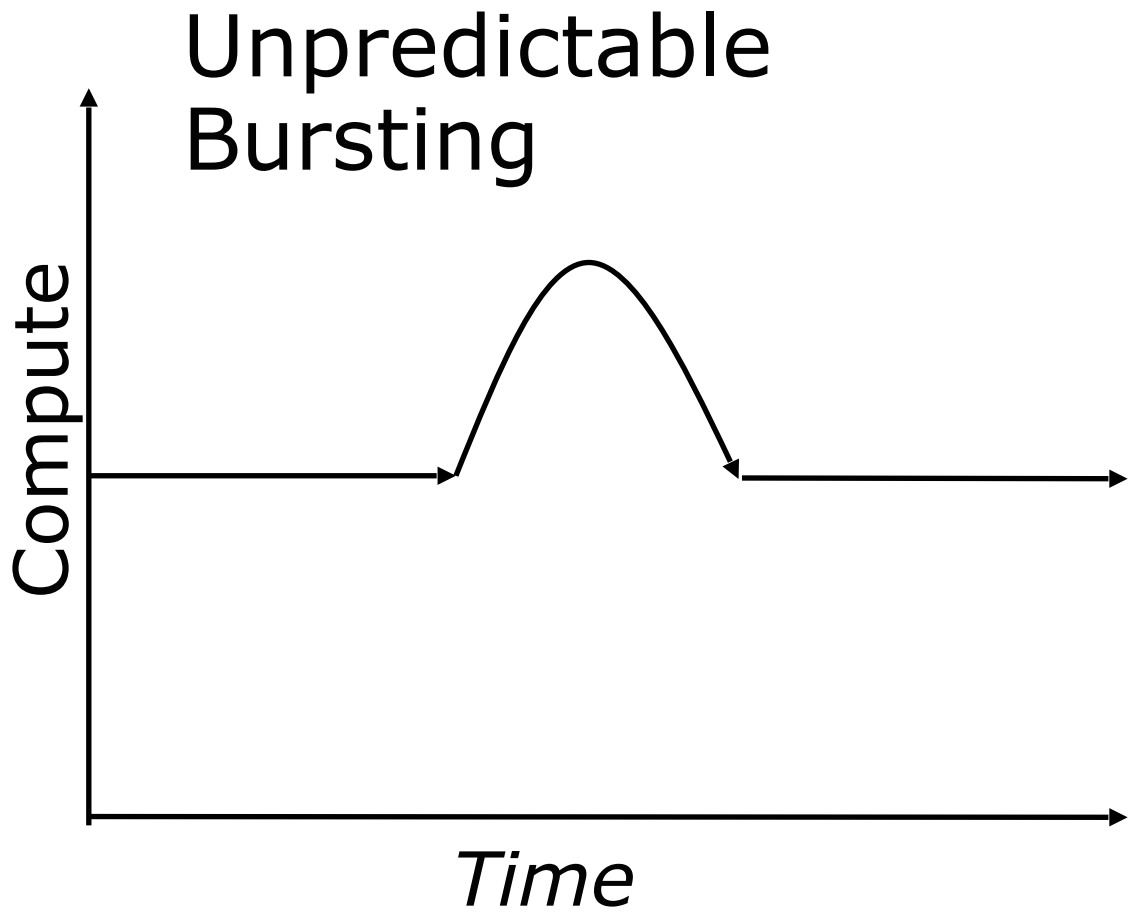
Typical Computing Pattern

- Rapidly growing company
- Major challenge for IT dept. to keep up with growth
- Potential loss of business opportunity
- Potential customer service problems



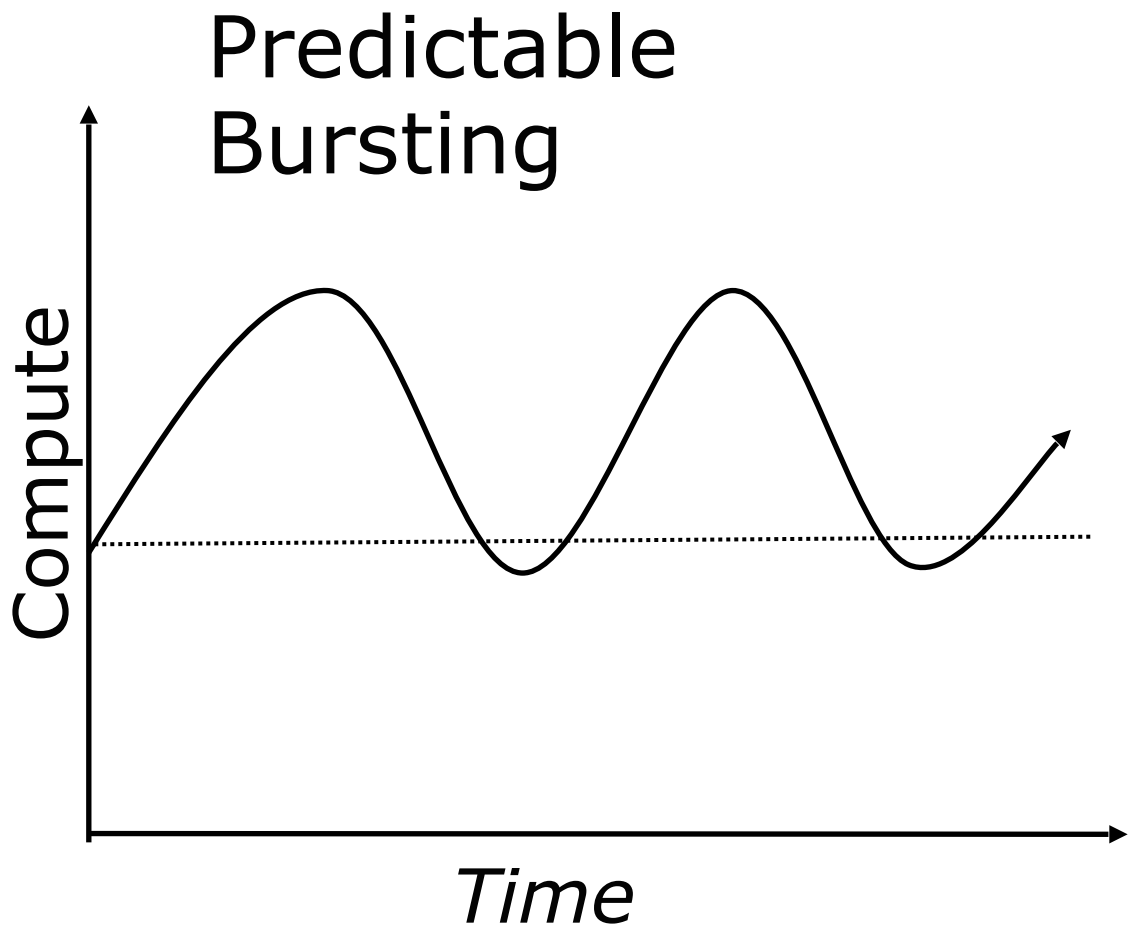
Typical Computing Pattern

- Unexpected peak in demand
- Loss of business opportunity
- Wasted capacity if demand wanes

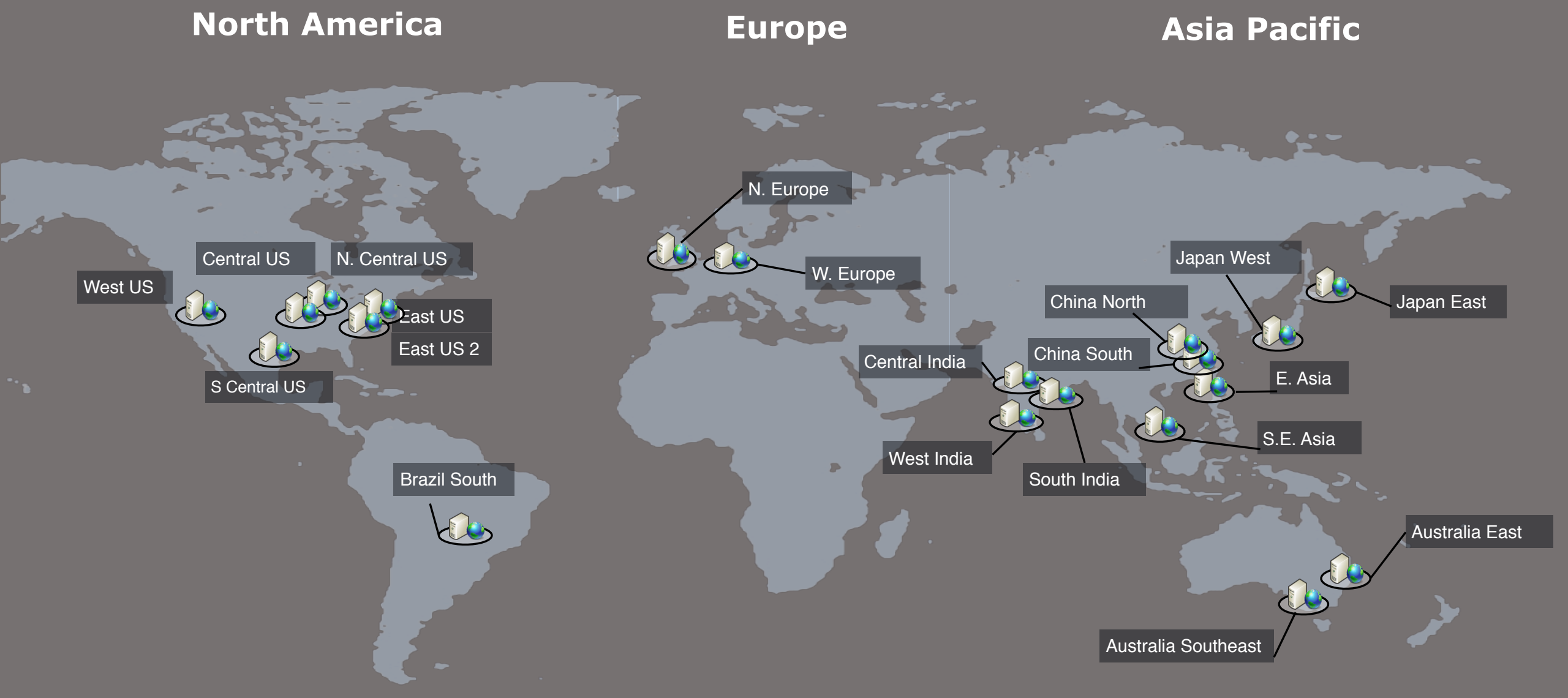


Typical Computing Pattern

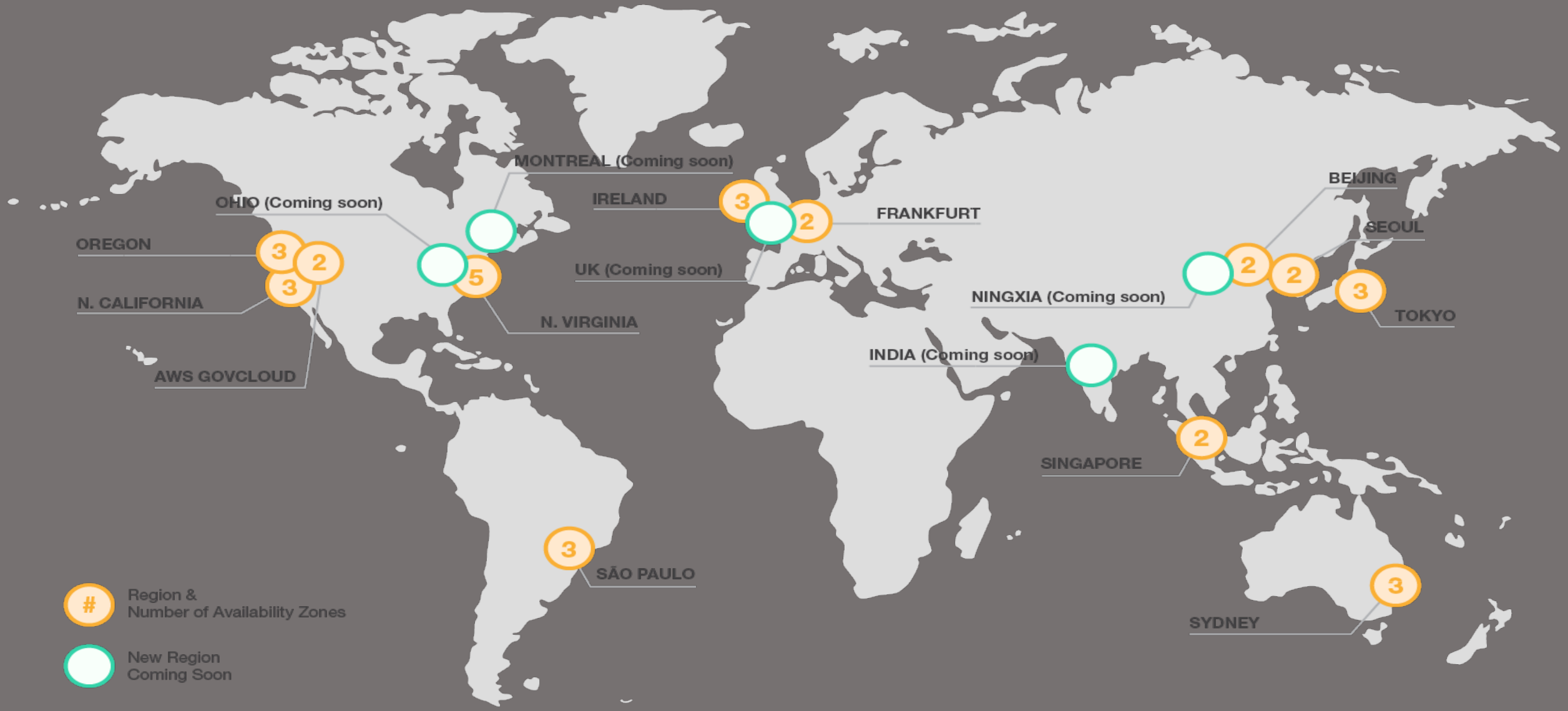
- Seasonal peaks and troughs
- Provisioning dilemma
  - Wasted capacity or
  - Loss of business



Azure Datacenter Regions



Amazon AWS Datacenter Regions

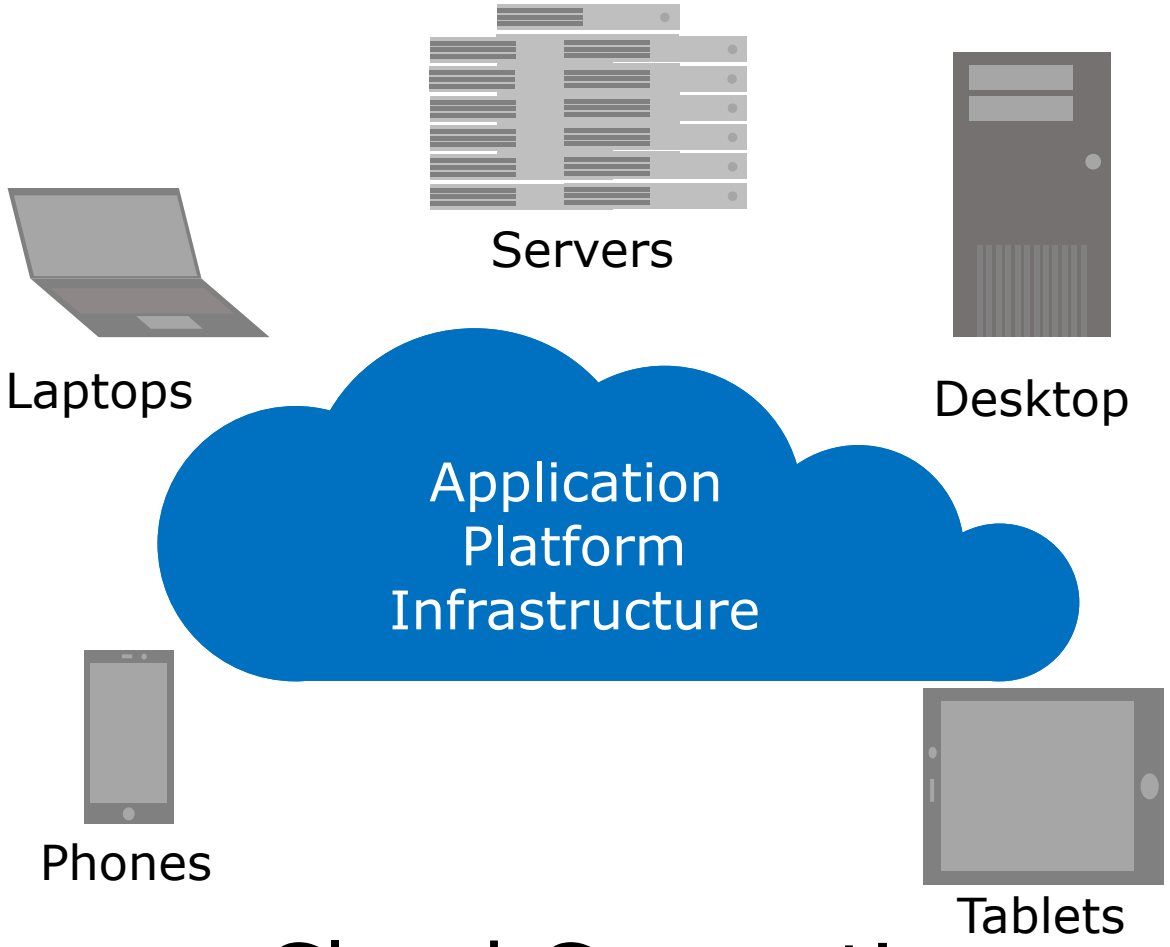


Cloud Computing Examples

- A large enterprise quickly & economically deploys new internal applications to its distributed workforce.
- An e-commerce website accommodates sudden demand for a “hot” product caused by a viral buzz.
- A pharmaceutical research firm executes large-scale simulations using computing power provided by cloud vendors.
- A media company serves unlimited video, music, and other media to their worldwide customer base.

Cloud Computing Nutshell

- End-users connect over the Internet to the cloud from their own personal computers or portable devices in order to access services.
- To the end-user, the underlying infrastructure such as the hardware, operating system, etc., is invisible



Cloud Computing

Cloud Vendor - Azure & AWS

Microsoft Azure and Amazon Web Services (AWS) offer broad and deep capabilities with global coverage

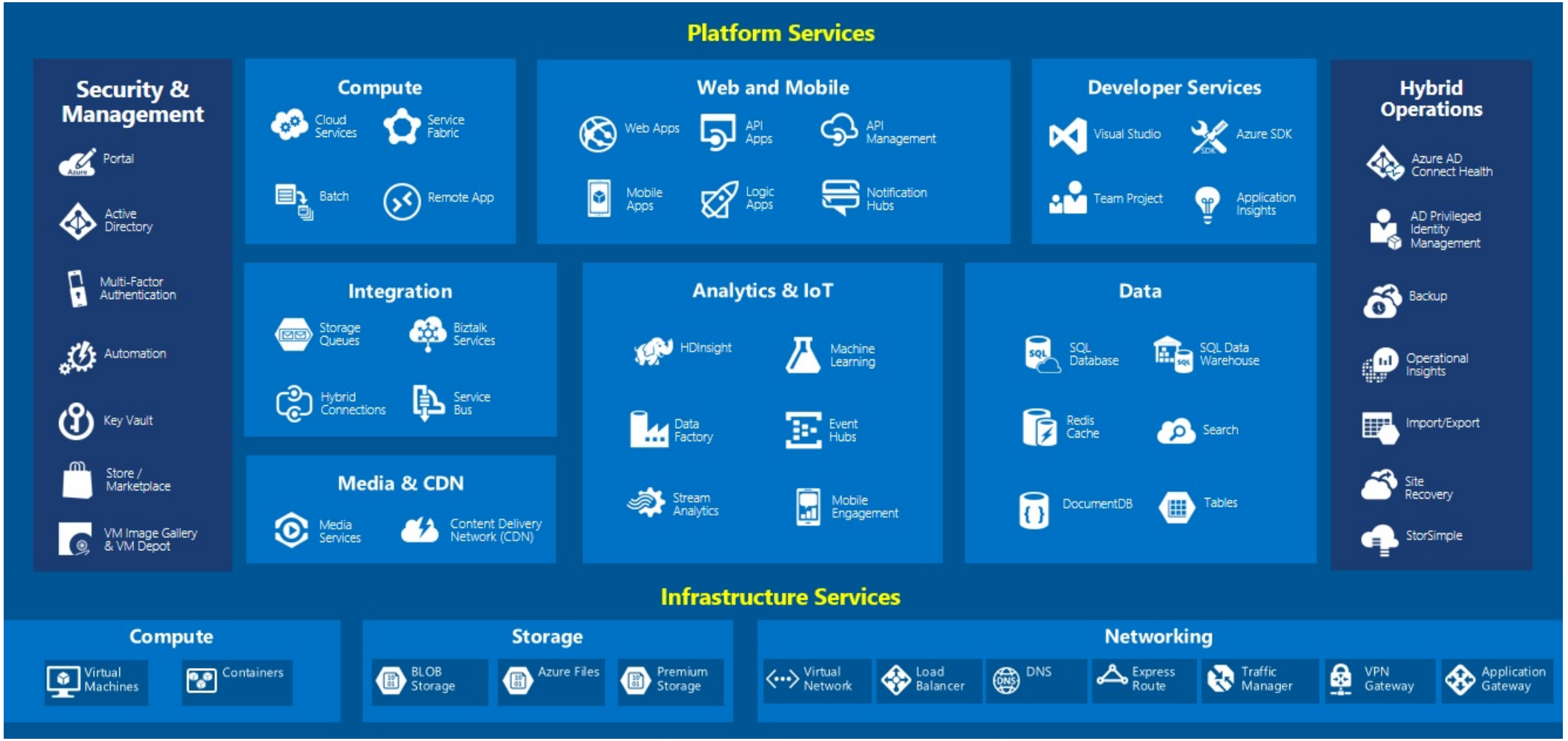
Category	Azure Service	AWS Service
Computing infrastructure	Virtual Machines	EC2
Object storage infrastructure	Blob Storage	S3
Networking	Virtual Network	Virtual Private Cloud
Relational database-as-a-service	SQL Database	RDS
NoSQL document database	DocumentDB	DynamoDB
Big data processing	HDInsight	Elastic MapReduce (EMR)
Visualization	Power BI	QuickSight

Cloud Vendor - Bluemix & Google

IBM Bluemix and Google Cloud each offer and deploy applications on highly-scalable and reliable infrastructure

Category	Bluemix	Google Service
Computing infrastructure	Virtual Server, Containers	Compute Engine
Object storage infrastructure	Object, Block Storage	Cloud Storage
Networking	Virtual Private Network	Cloud Virtual Network
Relational database-as-a-service	SQL Database	Cloud SQL
NoSQL document database	MongoDB	Cloud Datastore, Bigtable
Big data processing	Analytics for Apache Hadoop	BigQuery, Cloud Dataproc
Visualization		

Azure Services



Azure Usage

- Azure Active Directory Users
  - More than 500 Million
- Storage transactions per day
  - More than 777 Trillion
- Messages processed by Azure IoT per month
  - More than 1.5 Trillion
- Active Websites
  - More than 250,000

- Percentage of Fortune 500 Companies using Azure
  - More than 80%
- Authentications per week
  - More than 13 Billion
- SQL Databases in Azure
  - More than 1.5 Million
- Developers registered with Visual Studio Online
  - More than 1 million

Vendor Lock-In

Companies that adopt cloud computing must be wary of potential vendor lock-in issues

- Company’s entire data is stored with a single vendor’s cloud storage
- Company relies on a single vendor for all of its computations
- Changing vendors can be very costly

Survey of Cloud Computing and Azure Foundation

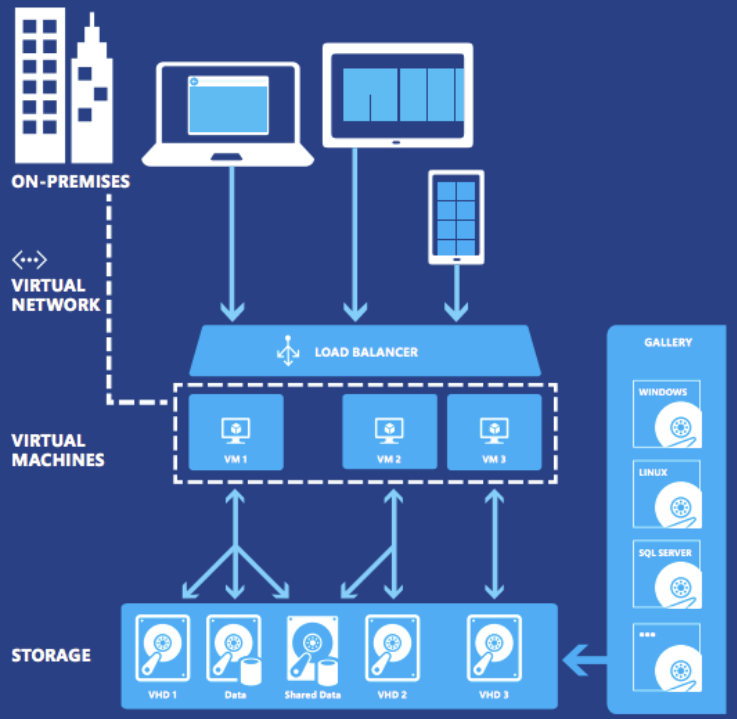
Services in Azure

What is Microsoft Azure?

Azure is a flexible, open, and secure public and hybrid cloud

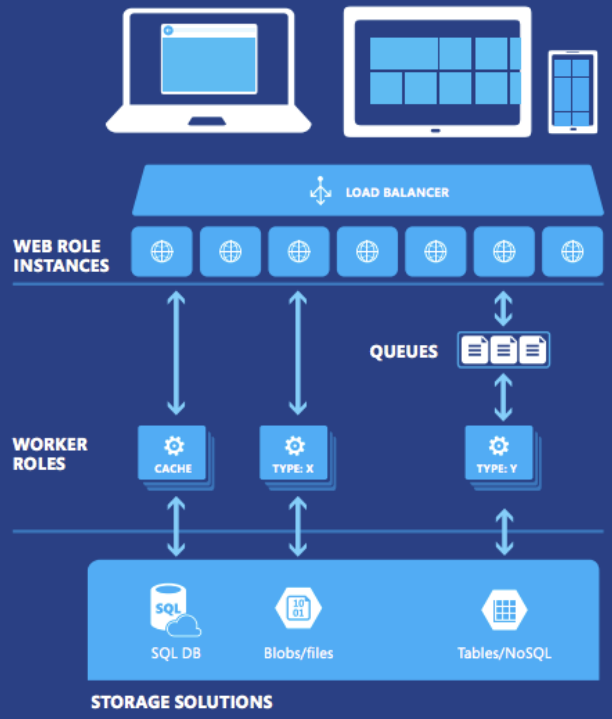
Virtual Machines

VMs are basic cloud building blocks. Get full control over a virtual machine with virtual hard disks. Install and run software yourself. Configure multiple machines with different roles to create complex solutions. VMs are nearly identical to conventional (real) servers, and are the easiest way to move existing workloads to the cloud.



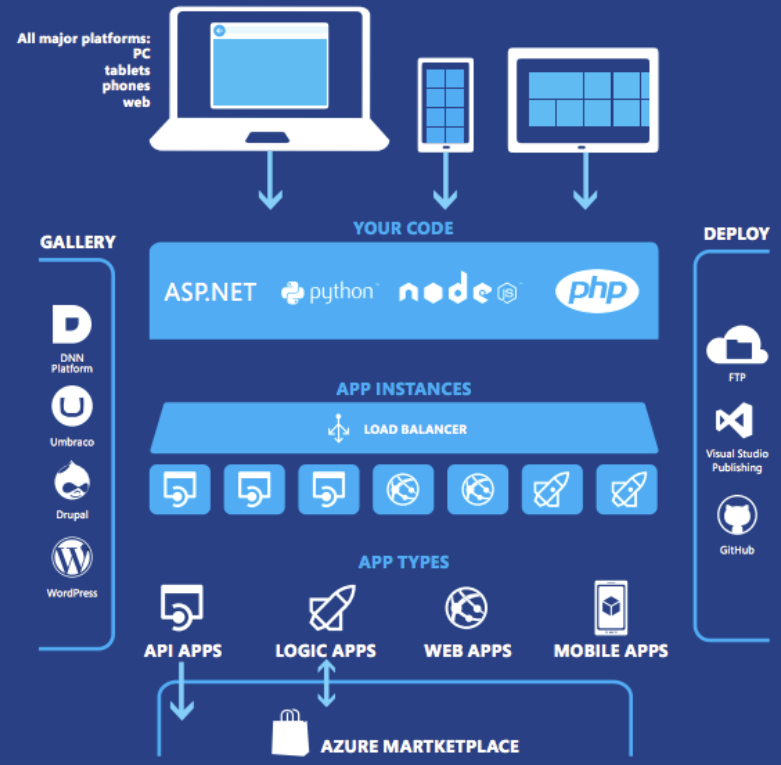
Cloud Services

Easily access and manage these general-purpose VMs. We maintain and update each VM as needed with system updates. You configure the VM size as needed, and scale out as many copies as needed. Two types of VMs: worker roles and web roles—worker roles are made for computing and running services. The web role is simply a worker role with IIS already installed and configured.



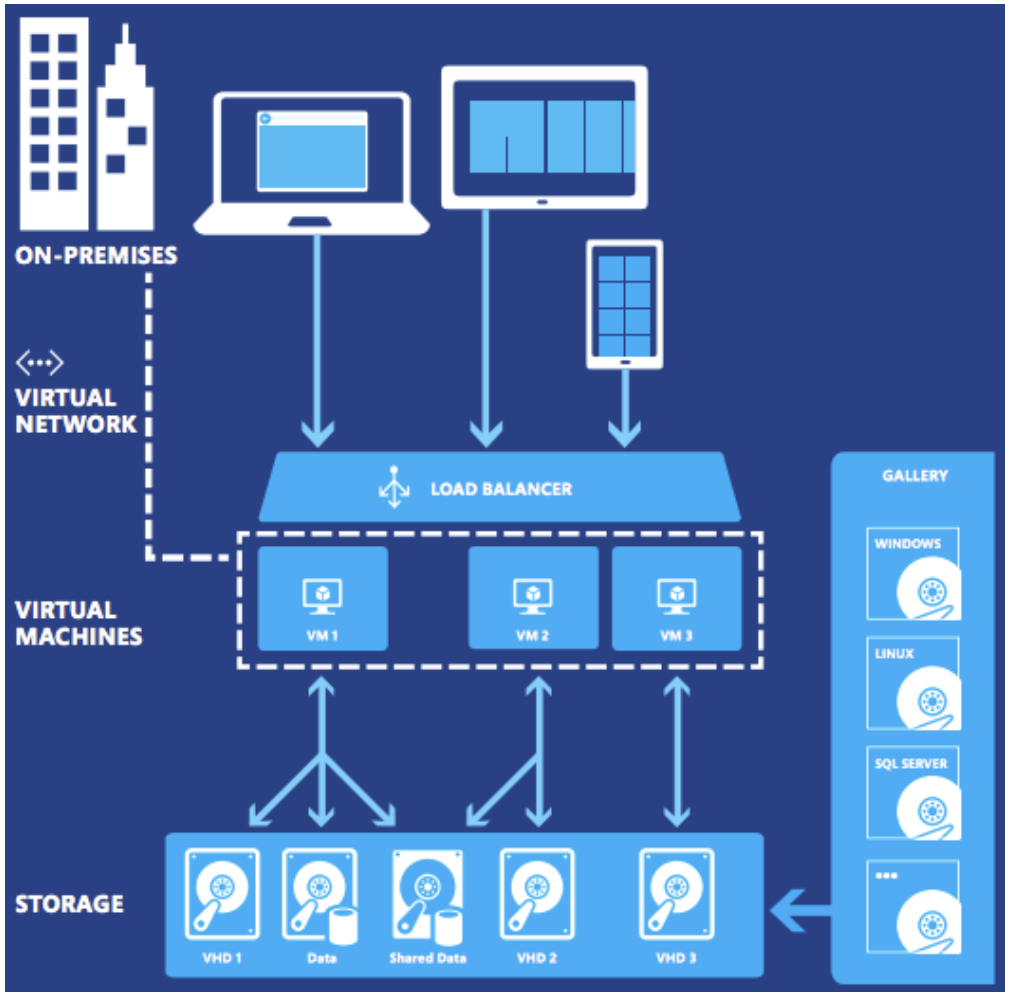
App Service

Azure App Service is a high productivity solution for developers who need to create enterprise-grade web and mobile app experiences. App Service provides a complete platform as a service solution that enables you to deploy and elastically scale applications in the cloud, and seamlessly integrate them with on-premises resources and SaaS based applications.



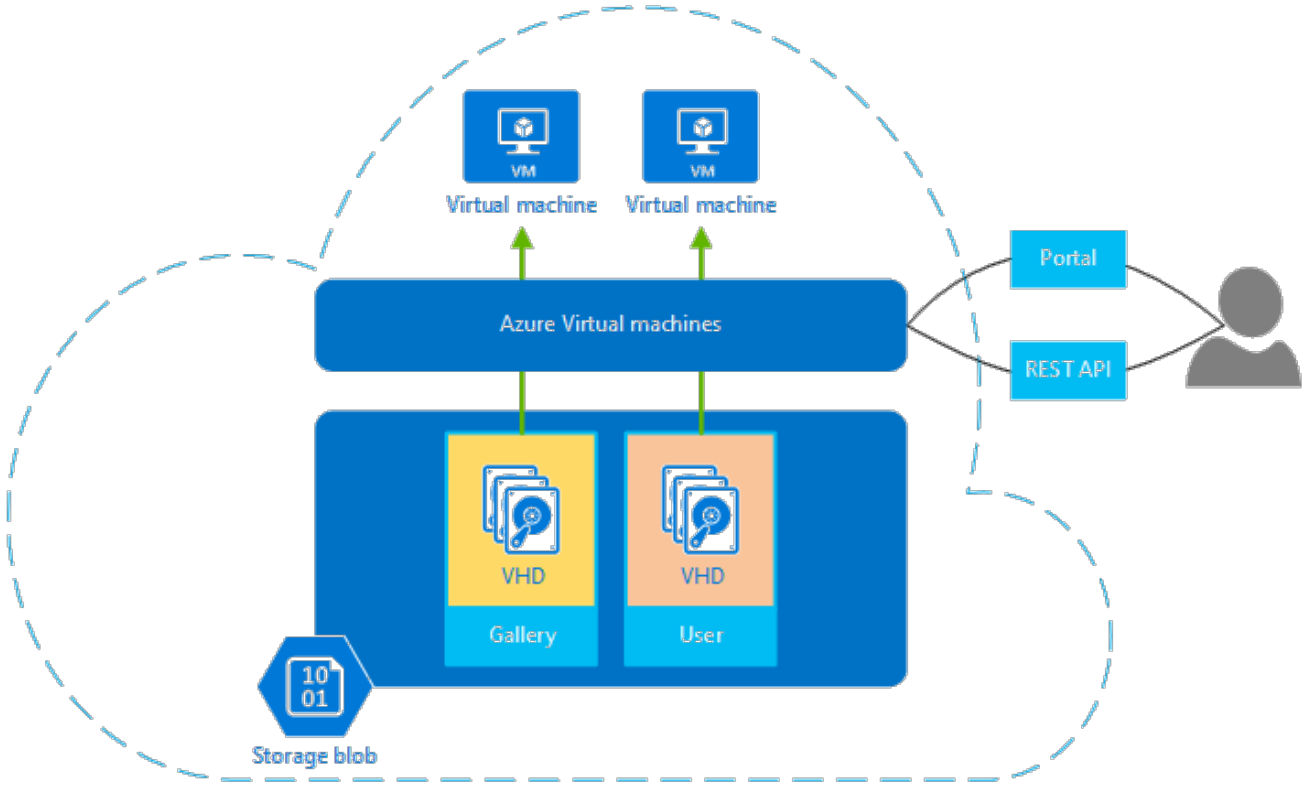
Virtual Machines

- Basic cloud building blocks
- Full control over virtual machines
- Install and run user applications
- Gallery of VM images available



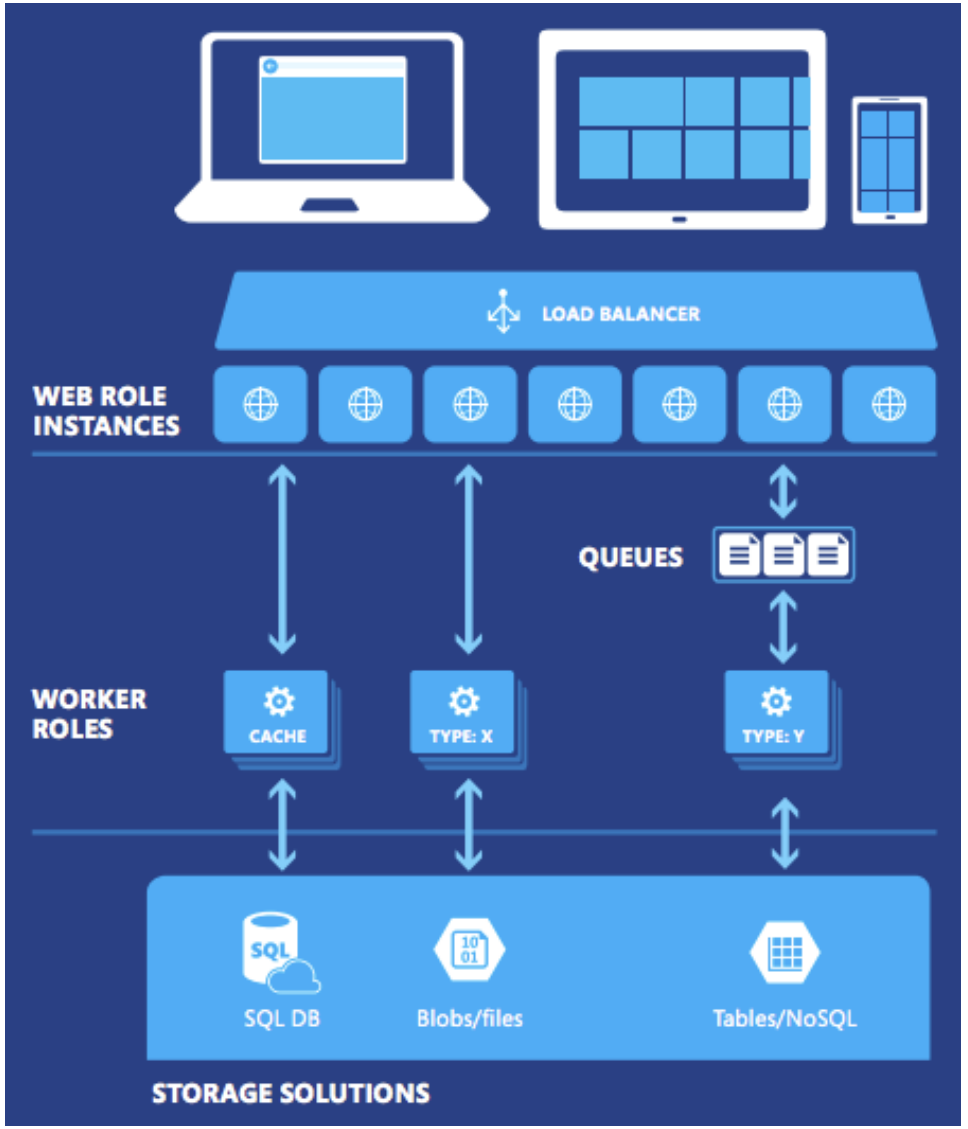
Azure Virtual Machines – IaaS

- IaaS Service
- VHD (Virtual Hard Disk) for persistent storage
- Different types of VMs can be configured together to provide complex solutions



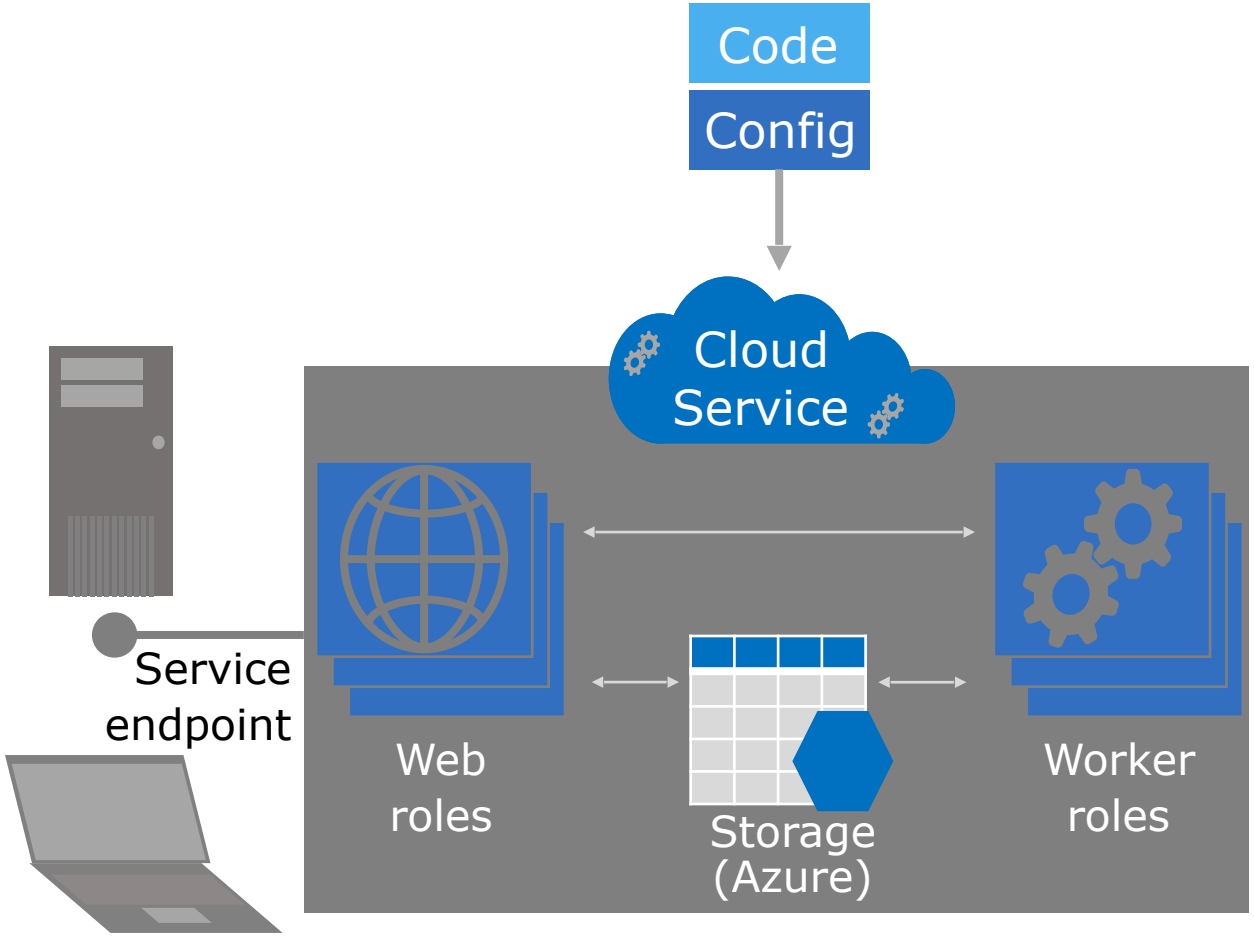
Cloud Service

- General purpose VMs available to user
- Users configure size of VM to fit their needs
- Vendor maintains VM with system update



Cloud Services - PaaS

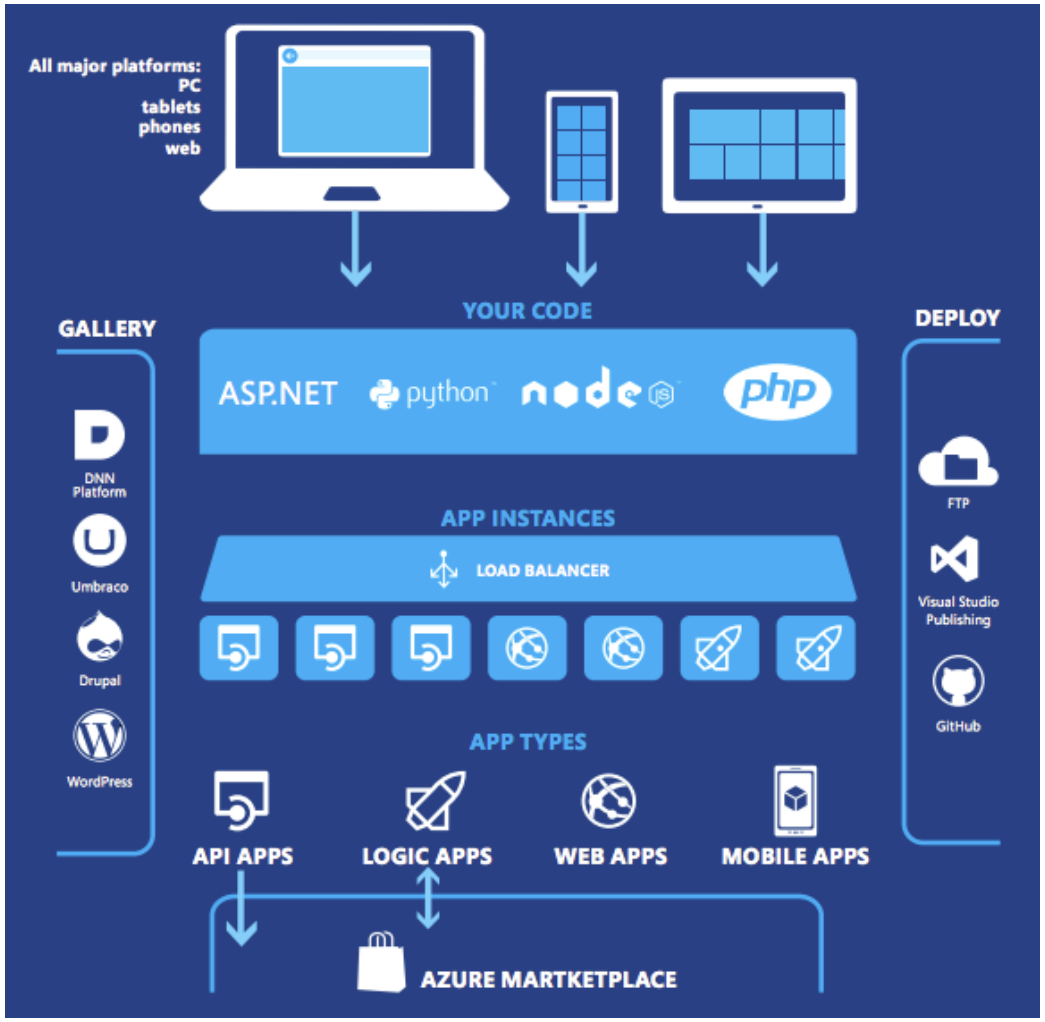
- PaaS service
- More flexibility over VMs than App Services
- Multiple worker role and/or web role VMs allowed for single application
- Single service access point for application end-user



App Service

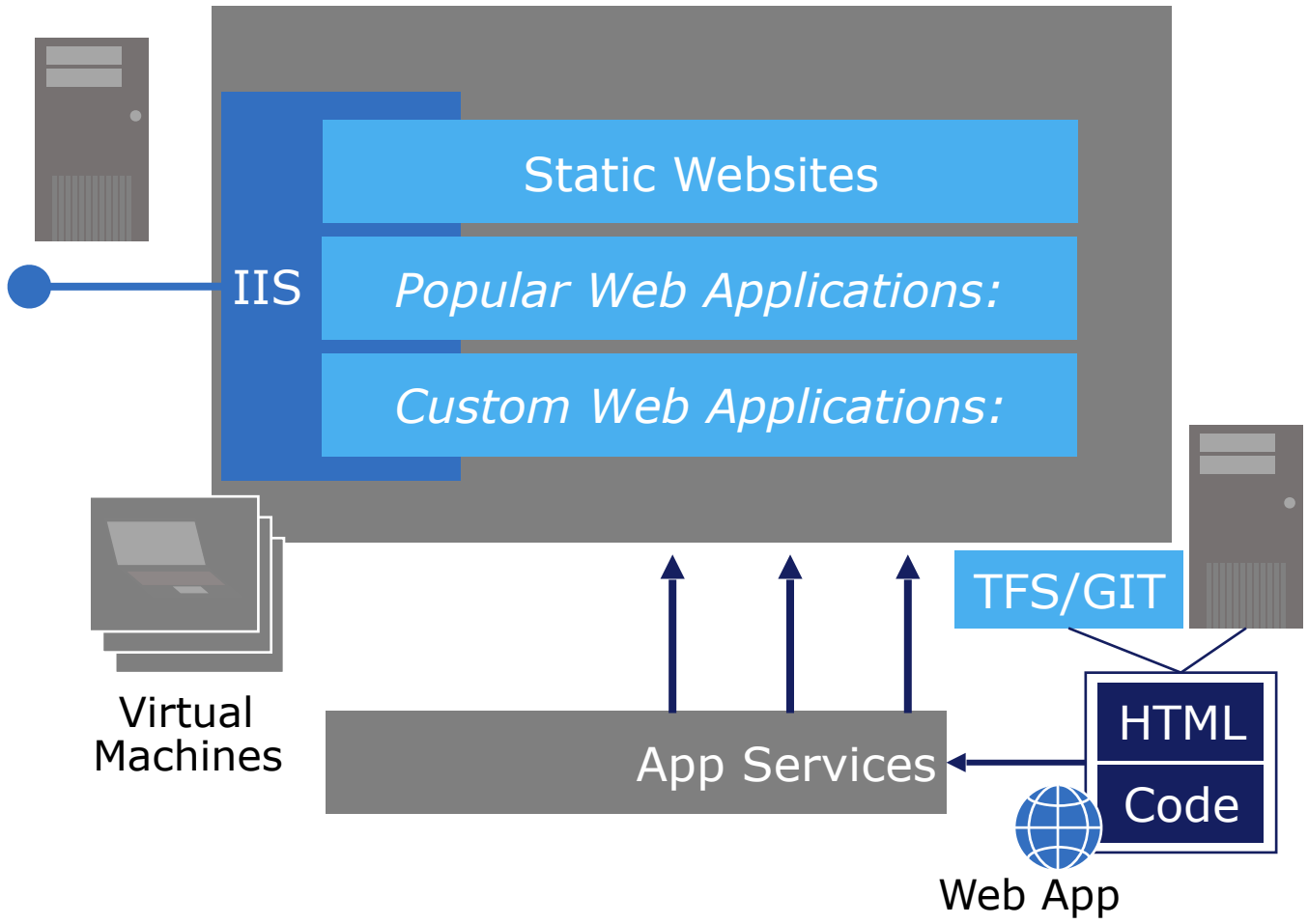
Features

- Multiple languages and frameworks
- Global scale with high availability
- Connections to SaaS platforms and on-premises data
- Security and compliance
- Application Templates
- DevOps optimization



App Service - PaaS

- PaaS service
- Optimal for quick web and mobile apps development
- Popular web apps – Drupal, Wordpress, etc.
- Custom web apps – ASP.NET, PHP, Node.js, etc
- Integration with SAP, Oracle EBS, SQL Server, and PeopleSoft via BizTalk Services



Why App Service - Web Apps?

- Services optimized to rapidly build, deploy and manage enterprise grade websites
- Jump-start using large list of existing templates
- Automatically scale depending on usage and traffic
- Integration and versioning with Visual Studio Team system, Github, TeamCity, Hudson or BitBucket
- Tools for staged deployment and verification
- Easy connection and access to on-premises data

Web Apps

Other Options

Azure also offers other compute hosting models for more specialized purposes

Model	Description
Mobile	Optimized to provide a cloud back-end for apps that run on mobile devices.
Batch	Optimized for processing large volumes of similar tasks, ideally workloads which lend themselves to running as parallel tasks on multiple computers.
HDInsight (Hadoop)	Optimized for running MapReduce jobs on Hadoop

Azure Service Modules - 1

Azure provides a broad collection of integrated services

Compute	Web and Mobile	Data and Storage	Analytics
Virtual Machines Cloud Services Batch RemoteApp Service Fabric Azure Container Service	App Service Logic Apps Web Apps Mobile Services API Management Mobile Engagement	DocumentDB SQL Database Redis Cache Storage StorSimple Search SQL Data Warehouse SQL Server Stretch Database	HDInsight Machine Learning Data Factory Data Catalog Data Lake Store Data Lake Analytics

Azure Service Modules - 2

Azure provides a broad collection of integrated services

Networking	Media and CDN	Hybrid Integration	Identity and Access Management
Virtual Network ExpressRoute Application Gateway Traffic Manager Azure DNS Load Balancer VPN Gateway	Media Services Content Delivery Network	BizTalk Services Service Bus Backup Site Recovery	Azure Active Directory Multi-Factor Authentication Azure Active Directory Domain Services Azure Active Directory B2C

Azure Service Modules - 3

Azure provides a broad collection of integrated services

Developer	Management	Intelligence	Internet of Things
Visual Studio Application Insights Azure DevTest Labs	Key Vault Scheduler Automation Log Analytics Security Center	Cognitive Services	Machine Learning Stream Analytics Push Notification Event Hubs Internet of Things Azure IoT Hub

Popular Products in Azure

Virtual Machines

Provision Windows and Linux Virtual Machines in minutes

App Service

Create web and mobile apps for any platform and any device

SQL Database

Managed relational SQL Database-as-a-service

Storage

Durable, highly available and massively scalable cloud storage

Cloud Services

Create highly available, infinitely scalable cloud applications and APIs

DocumentDB

Managed NoSQL document database-as-a-service

Azure Active Directory

Synchronize on-premises directories and enable single sign-on

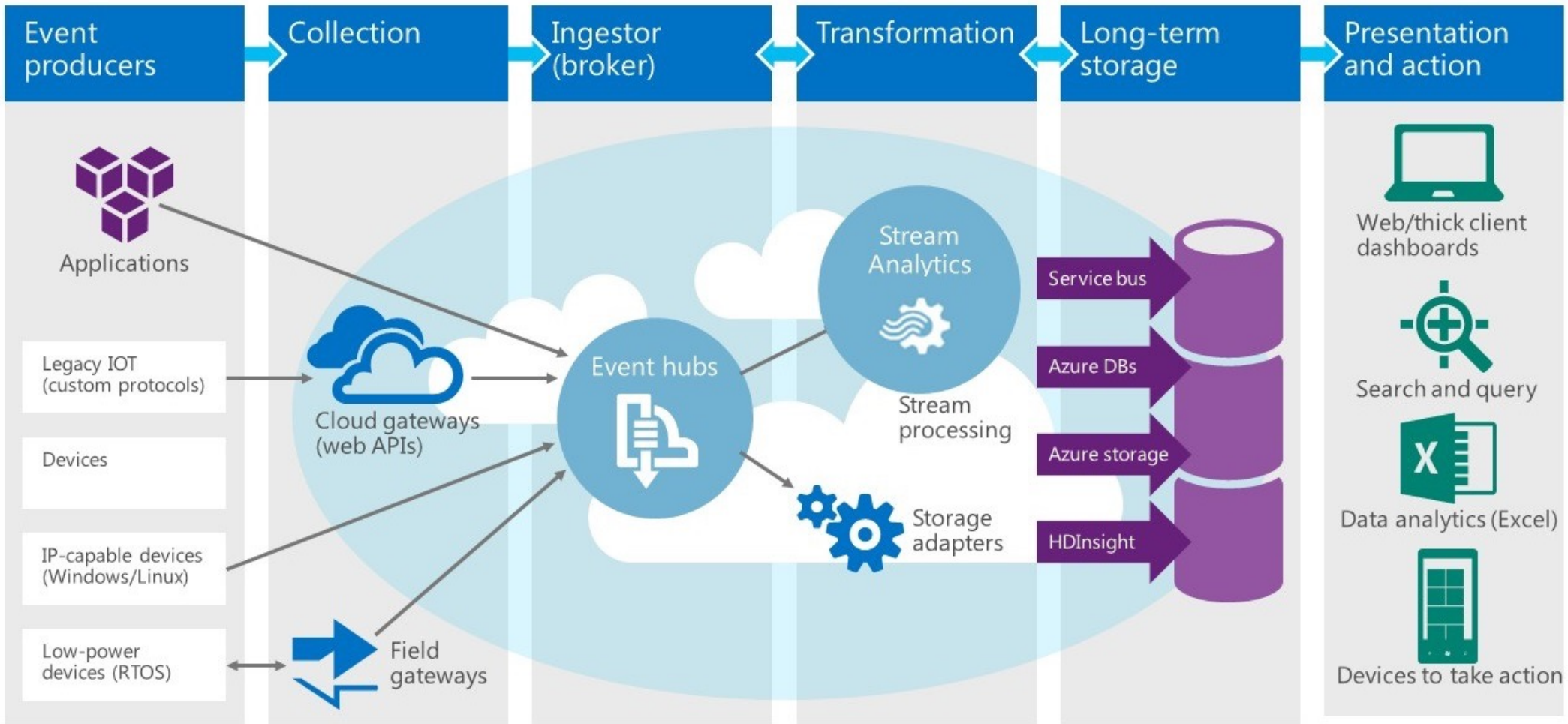
Backup

Simple and reliable server backup to the cloud

NoSQL vs. SQL DB

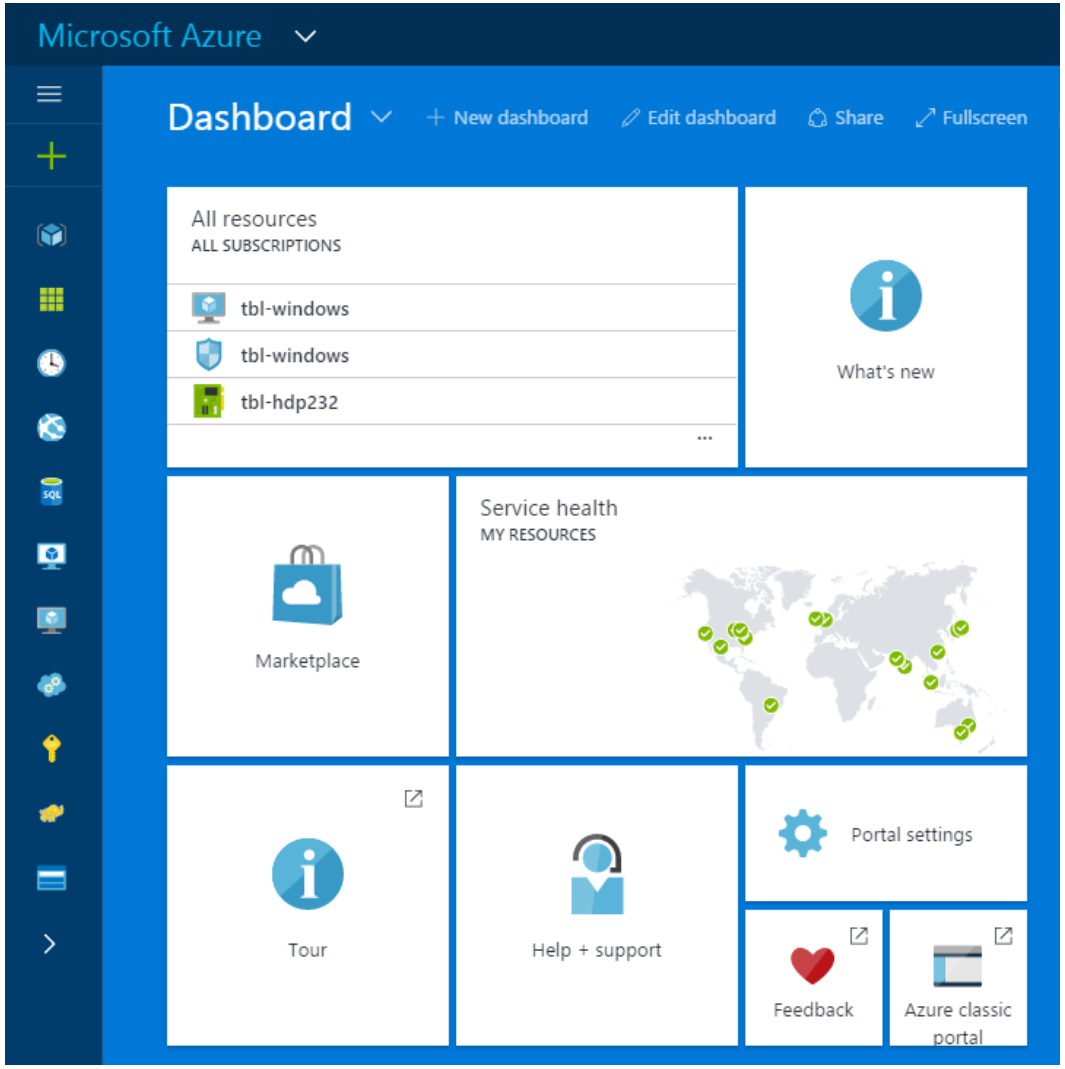
	NoSQL	SQL
Model	Non-relational – Stores data in JSON documents, key/value pairs, wide column stores, or graphs	Relational – Stores data in a table
Data	Offers flexibility as not every record needs to store the same properties. Good for semi-structured data.	Great for solutions where every record has the same properties. Good for structured data.
Schema	Dynamic or flexible schemas – database is schema-agnostic and the schema is dictated by the application. This allows for agility and highly iterative development.	Strict schema – Schema must be maintained and kept in sync between application and database
Transactions	ACID transaction support varies per solution.	Supports ACID transactions.
Consistency	Consistency varies per solution, some solutions have tunable consistency	Strong consistency supported.

Azure Services Work Together



Access Services Through Azure Portal

- The Microsoft Azure portal is the central location where Azure resources and services are provisioned and managed
- Marketplace for solutions
- Customized start screen



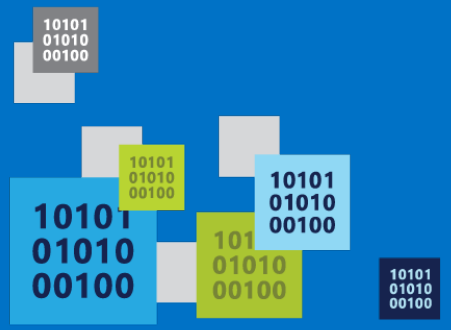
Role-Based Access Control in Azure Portal

Azure Role-Based Access Control (RBAC) enables fine-grained access management for Azure.

- Built-in roles
  - Owner has full access to all resources including the right to delegate access to others.
  - Contributor can create and manage all types of Azure resources but can’t grant access to others.
  - Reader can view existing Azure resources.

Microsoft Azure

Building Cloud Solutions



Microsoft

Host your applications on Azure

Microsoft Azure

Infrastructure as a Service
IaaS

Platform as a Service
PaaS

Microsoft

Your service

- → Your application code
  - → Required infrastructure code

Automation

Innovation

Reduced cycle time

Elasticity

Versioning environments

Continuous Deployment

Speed to market

Agility

Continuous Improvements

Feed back

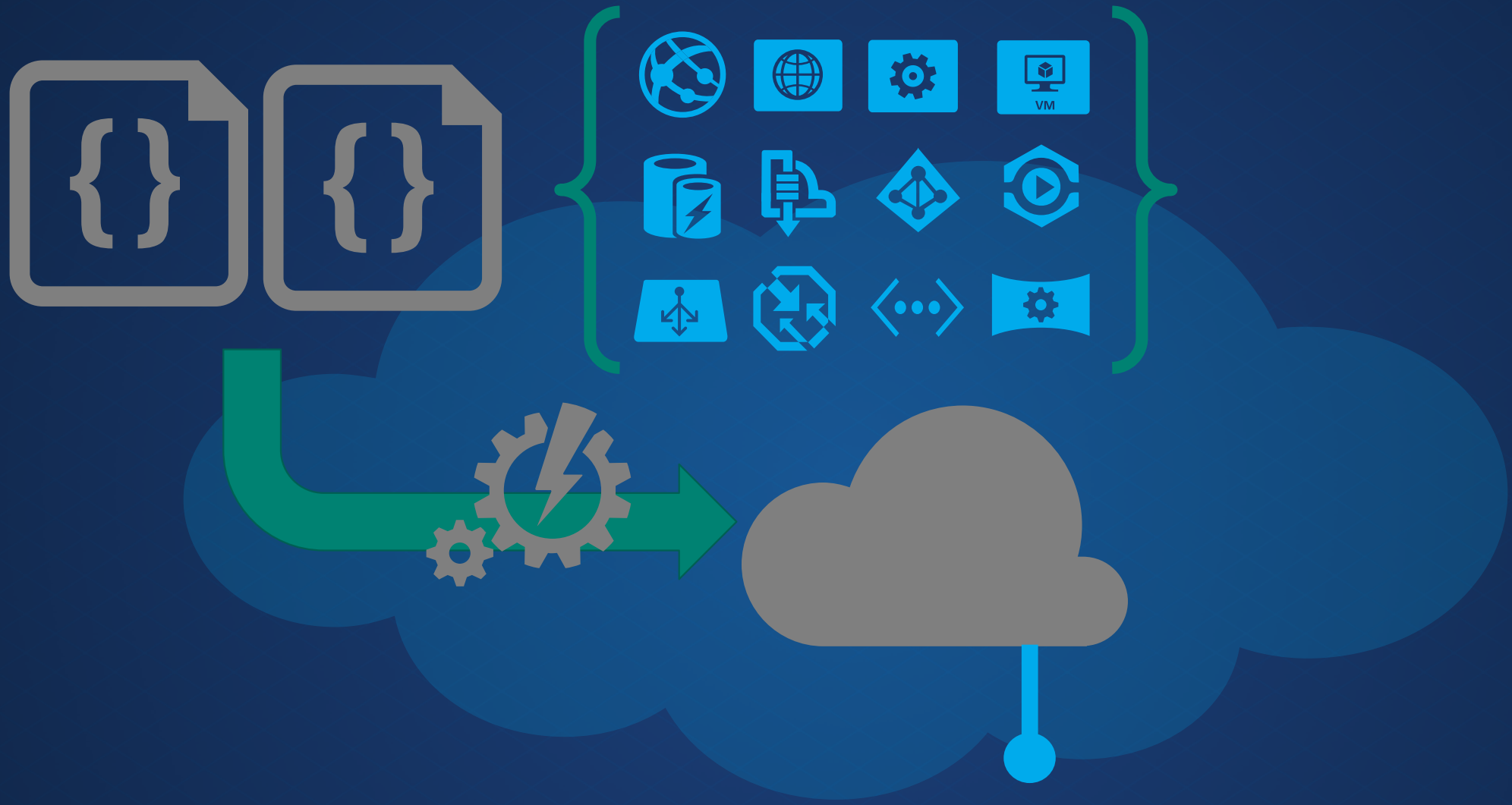
QoS

Availability

Growth

Insights

TCO



You: Code (application, infrastructure)
Azure: Resources (IaaS, PaaS, SaaS)

Microsoft Azure

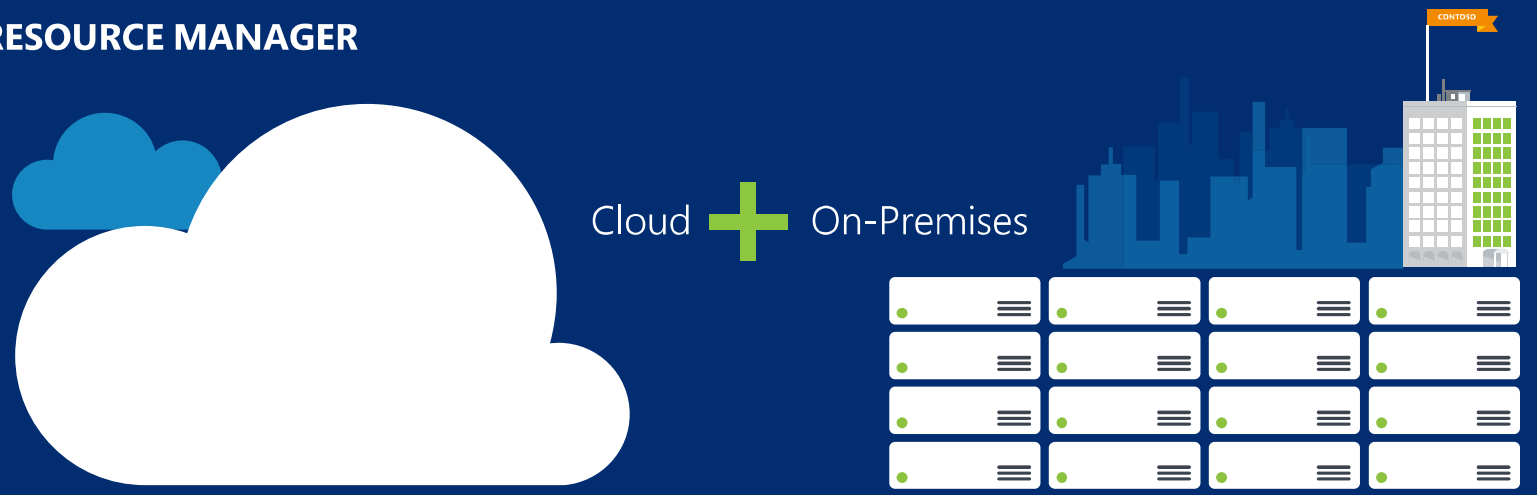
Consistent Management Layer

Tools

Microsoft Azure + Command Line + Visual Studio

SERVICE MANAGEMENT API

RESOURCE MANAGER



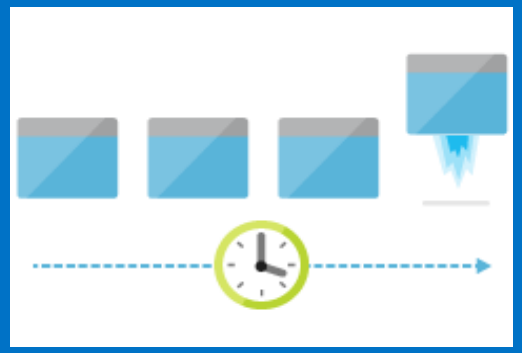
ADFS

AAD

RESOURCE PROVIDER CONTRACT

Provider Rest Points

Microsoft



Deploy

Organize

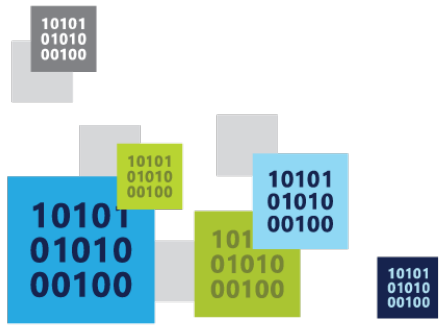
Control

- Deploy application lifecycle container with repeatable declarative model based template
- Organize resources by environment, role, department and user responsibility
- Control and monitor resources through RBAC, centralized audit and resource lock

Cloud Services

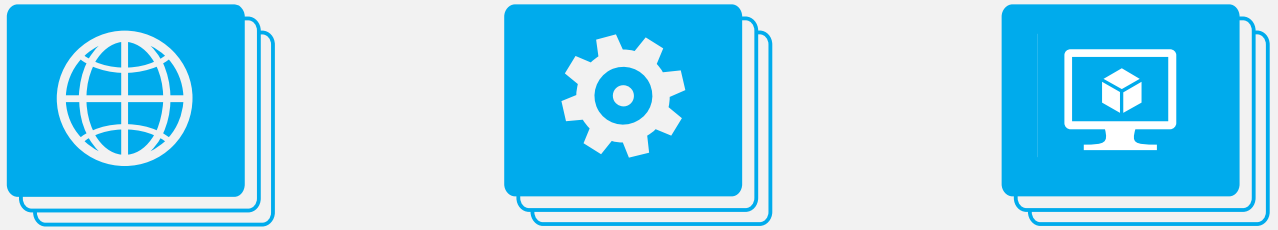
Cloud Services

- Focus on your application
- Scalability, availability and reliability
- Monitoring and diagnostics



What is a Cloud Service?

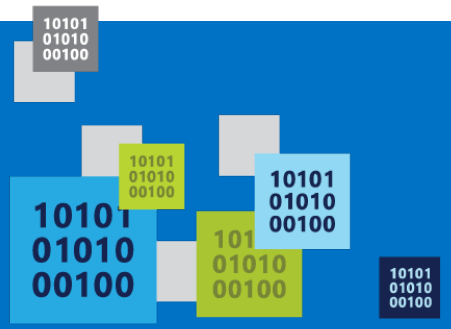
A container of related service roles



Web Roles

Worker Roles

VMs

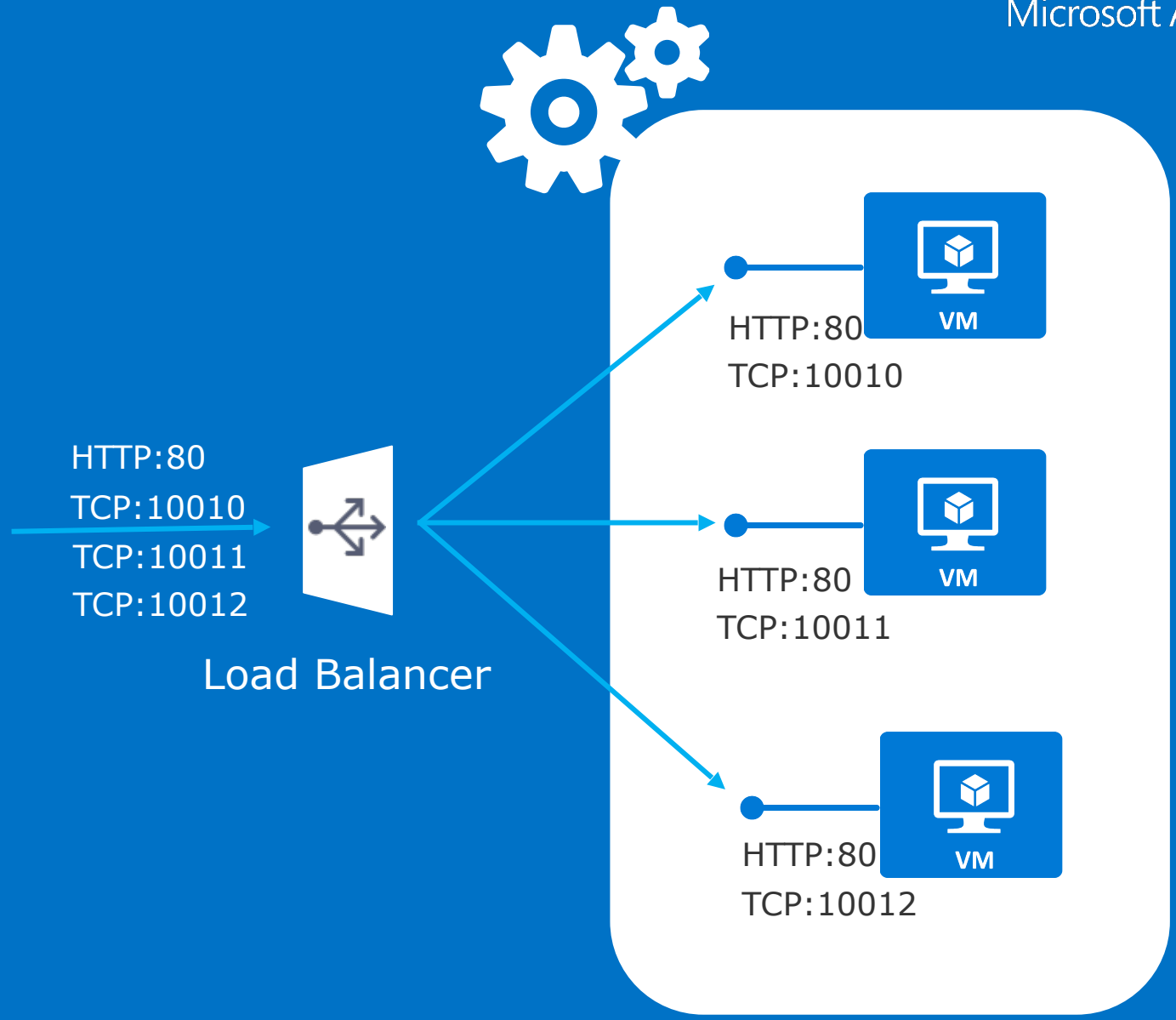


How do roles communicate?

- Public endpoints
  - Publicly accessible, load balanced
- Internal endpoints
  - Private to cloud service, not load balanced
- Instance Input endpoints
  - Address individual instance

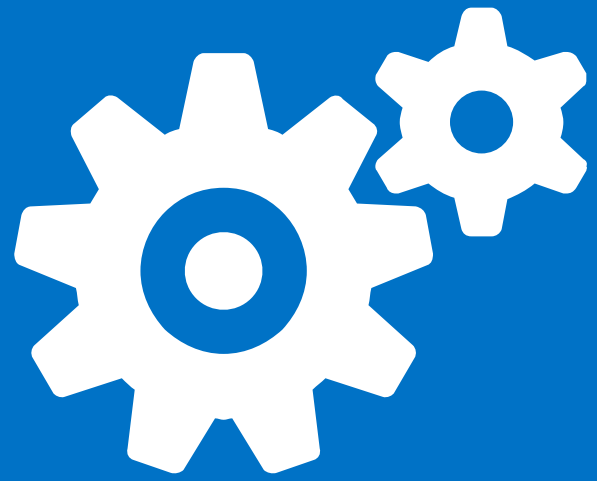
Endpoint

Name
Internal Port
Public Port
Protocol
IP



Worker Role

Microsoft Azure



Microsoft

Web Role

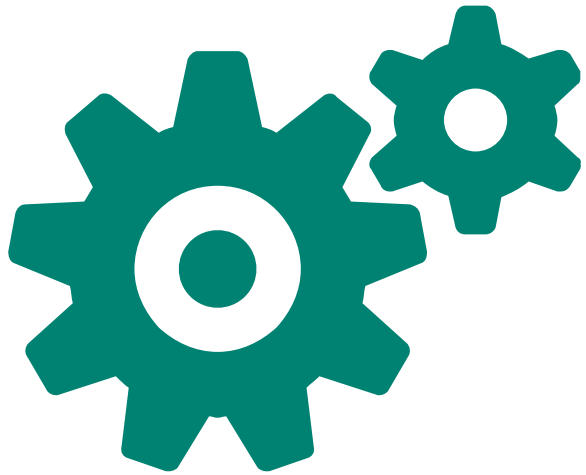
All features of a worker role + IIS 7, 7.5 or IIS 8.0*

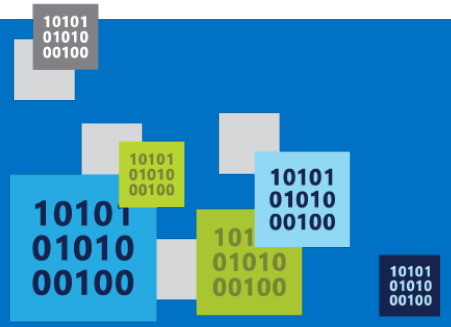
- All features of a worker role + IIS 7, 7.5 or IIS 8.0*
- ASP.NET 3.5 SP1, 4.0 or 4.5* – 64bit
- Hosts
  - Webforms or MVC
  - FastCGI applications (e.g. PHP)
  - Multiple Websites
- Http(s)
- Web/Worker Hybrid
  - Can optionally implement RoleEntryPoint

*with Windows Server 2012

Worker Role Patterns

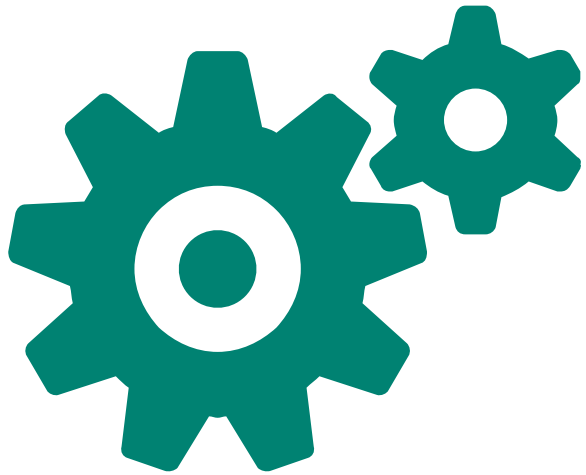
- Queue Polling Worker
  - Poll and Pop Messages within while(true) loop
  - E.g. Map/Reduce pattern, background image processing
- Listening Worker Role
  - Create TcpListener or WCF Service Host
  - E.g. Run a .NET SMTP server or WCF Service

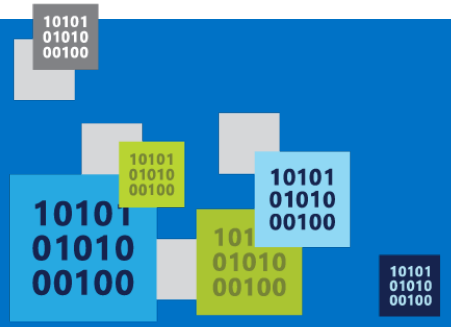




Worker Role Patterns

- External Process Worker Role
  - OnStart or Run method executes Process.Start()
  - Startup Task installs or executes background/ foreground process
- Custom Role Entry Point (executable or .Net assembly)
- E.g. Run a database server, web server, distributed cache





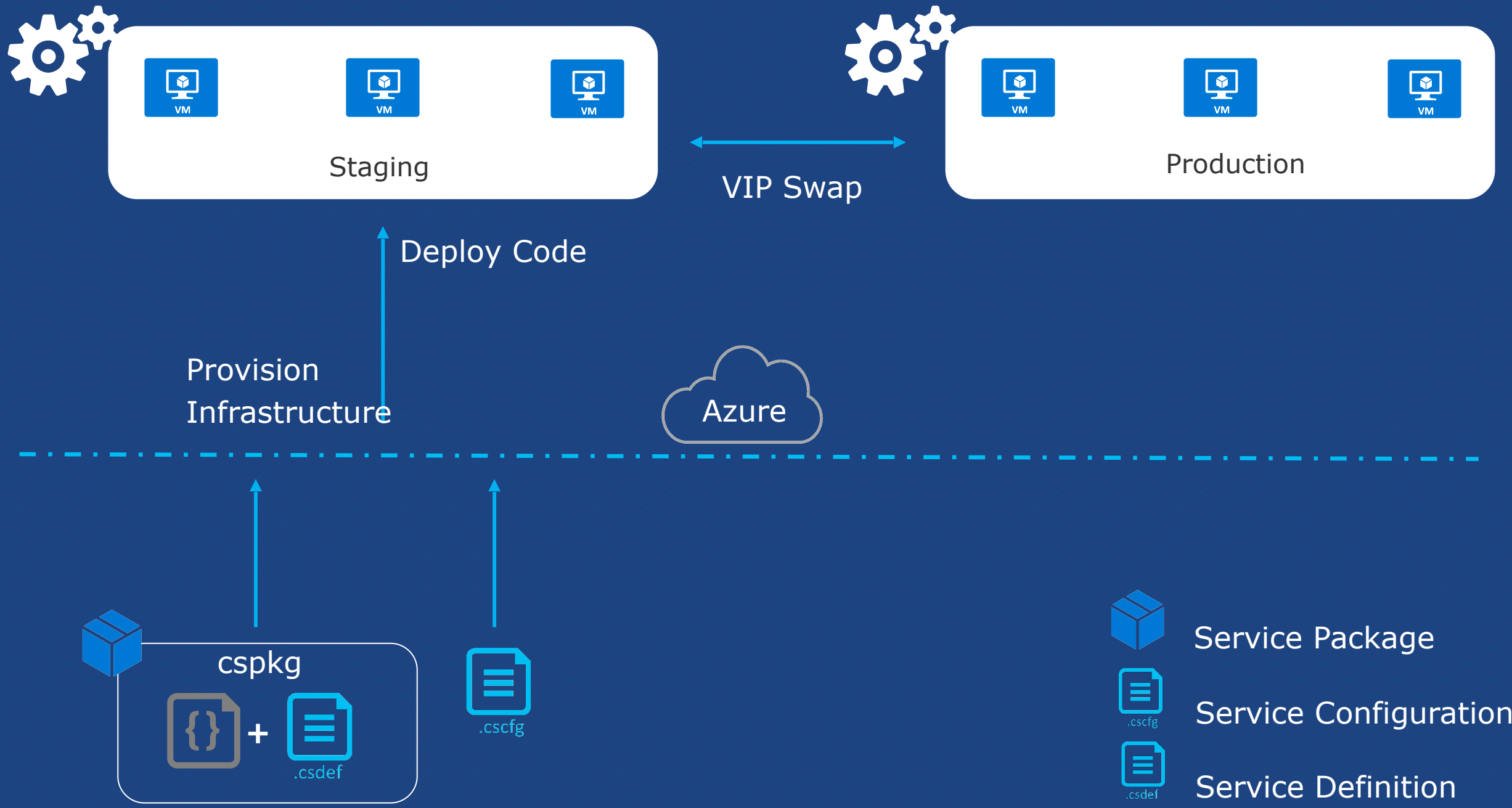
Roles and Instances

Roles are defined in a Hosted Service

A role definition specifies:
VM size
Communication Endpoints
Local storage resources
etc.

At runtime each Role will execute on one or more instances

A role instance is a set of code, configuration, and local data, deployed in a dedicated VM


VM
VM
VM
Staging
VIP Swap
VM
VM
VM
Production
Deploy Code
Provision
Infrastructure
Azure
cspkg
{}
+
.csdef
.cscfg
Service Package
.cscfg
Service Configuration
.csdef
Service Definition

Integrated development experience powered by Visual Studio + Azure SDK

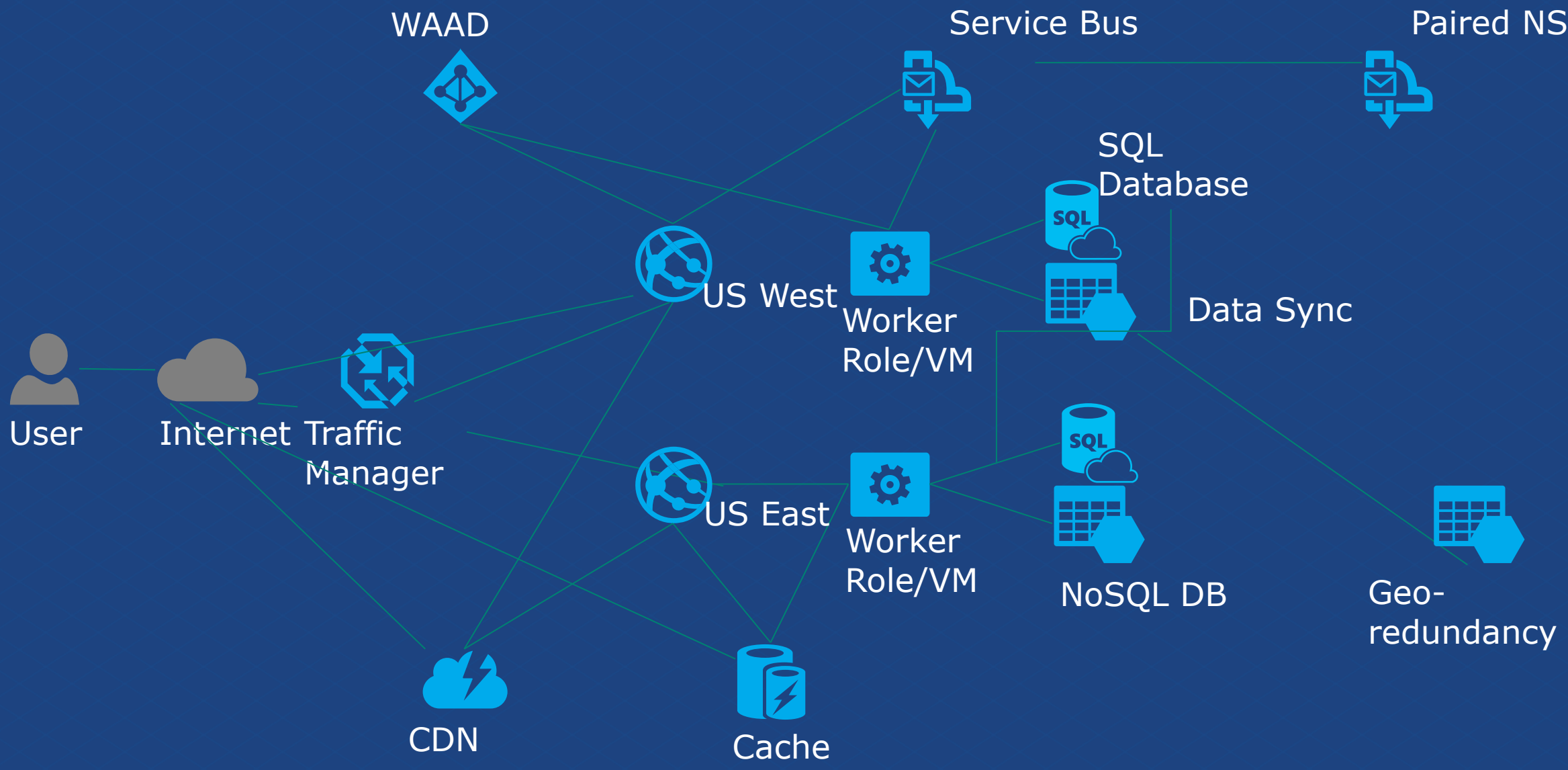
- Development experience using the Azure SDK, integrated seamlessly with Visual Studio.
- Deploy using any language you like, including .NET, Java, Node.js, PHP, Python, or Ruby.
- Test your application before deploying to the cloud using the Azure Emulator, which brings the platform's key functionality right to your dev machine.

Design for Cloud

A different mindset

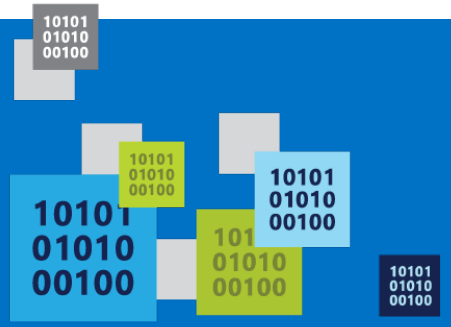
- → Embracing errors
- → Design for availability, reliability, scalability
- → Performance

Sample architecture



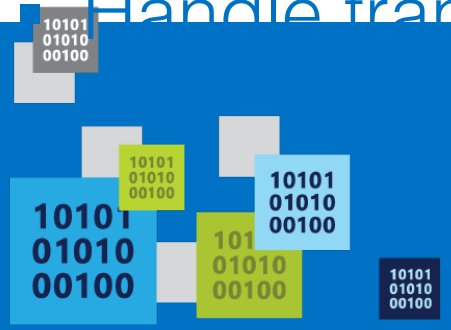
Redundancy in Microsoft Azure

- Storage Redundancy Options
- SQL Database Geo-Replication
- Caching with high availability option
- Load-balanced App Service Apps, Cloud Service and Virtual Machines
- Built-in redundancy in Azure Virtual Network gateways
- Failover with Azure Traffic Manager



Resiliency in Microsoft Azure

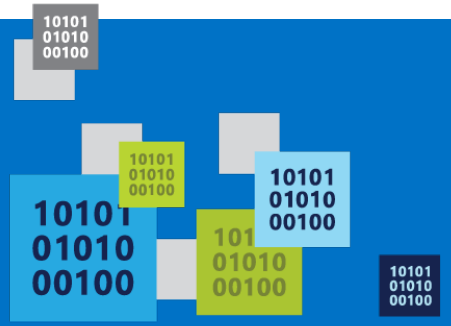
- Auto recovery – Service Healing
- Fault domain - prevent single point of failure
- Virtual machine Availability set – fault domain and rolling host updates
- Upgrade domain - service availability during upgrade
- Deployment Slots and VIP swap
- Emulator, Intellitrace and enhanced diagnostics
- Telemetry - native and 3rd party support
- Handle transient errors with Application Block



What does failsafe mean for my applications?

It depends… but some general practices apply.

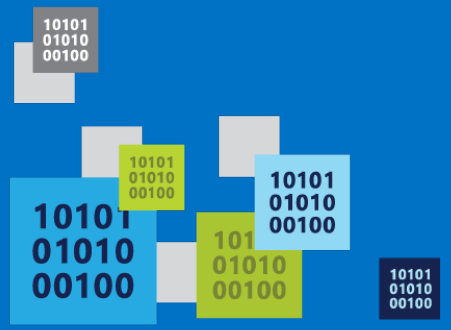
- Take advantage of Microsoft Azure features
- Avoid single point of failure
- Failure mode analysis
- Transient errors
- Graceful degradation
- Eliminate human factors



Scaling in Microsoft Azure

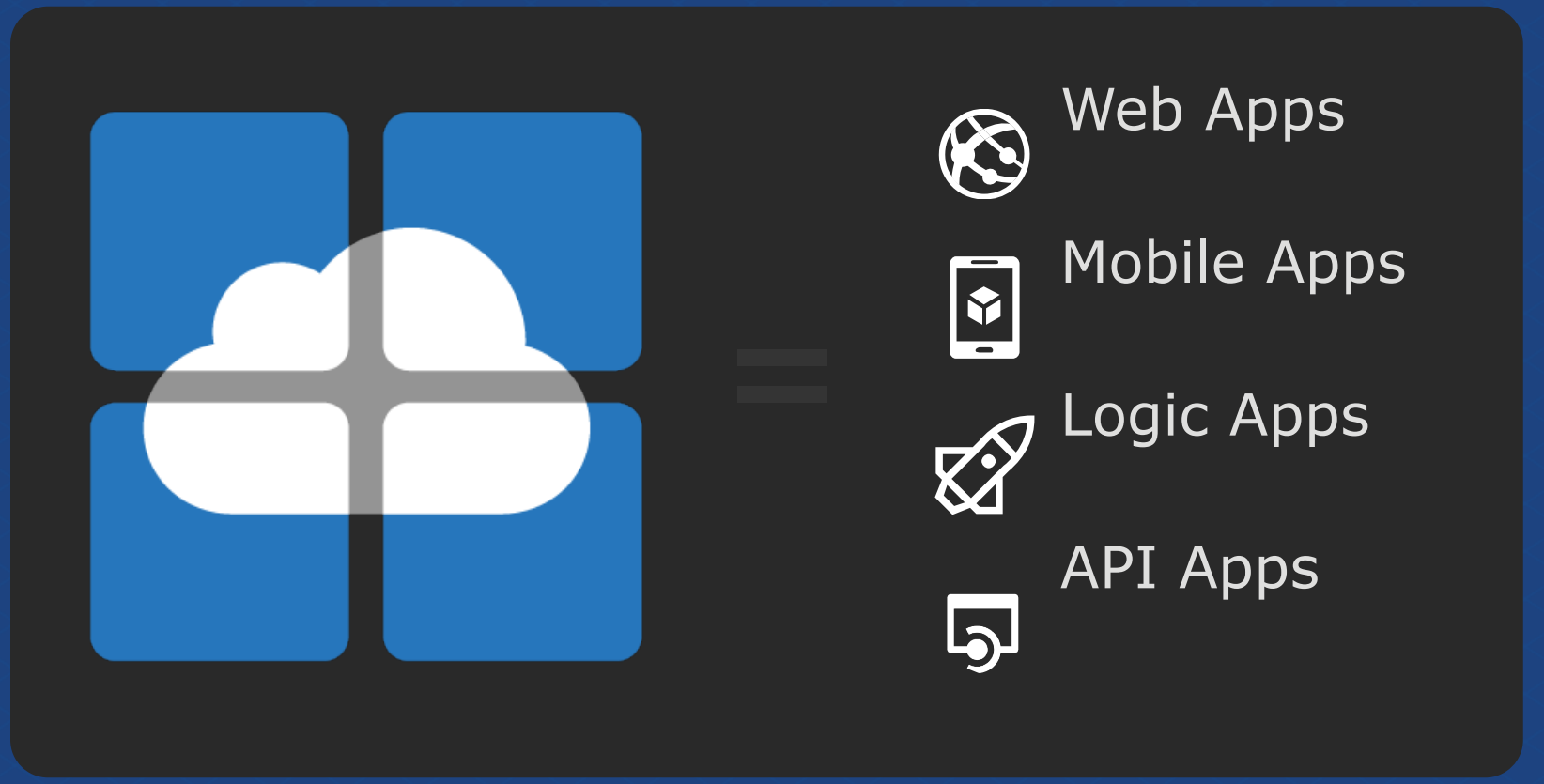
- Scale up by choosing different VM sizes
- Scale out by adding more instances
- Auto-scale with Autoscaling Application Block
- Scale out by using multiple service entities
- CDN to distribute user traffic
- Caching to offset server workloads

App Service Web Apps



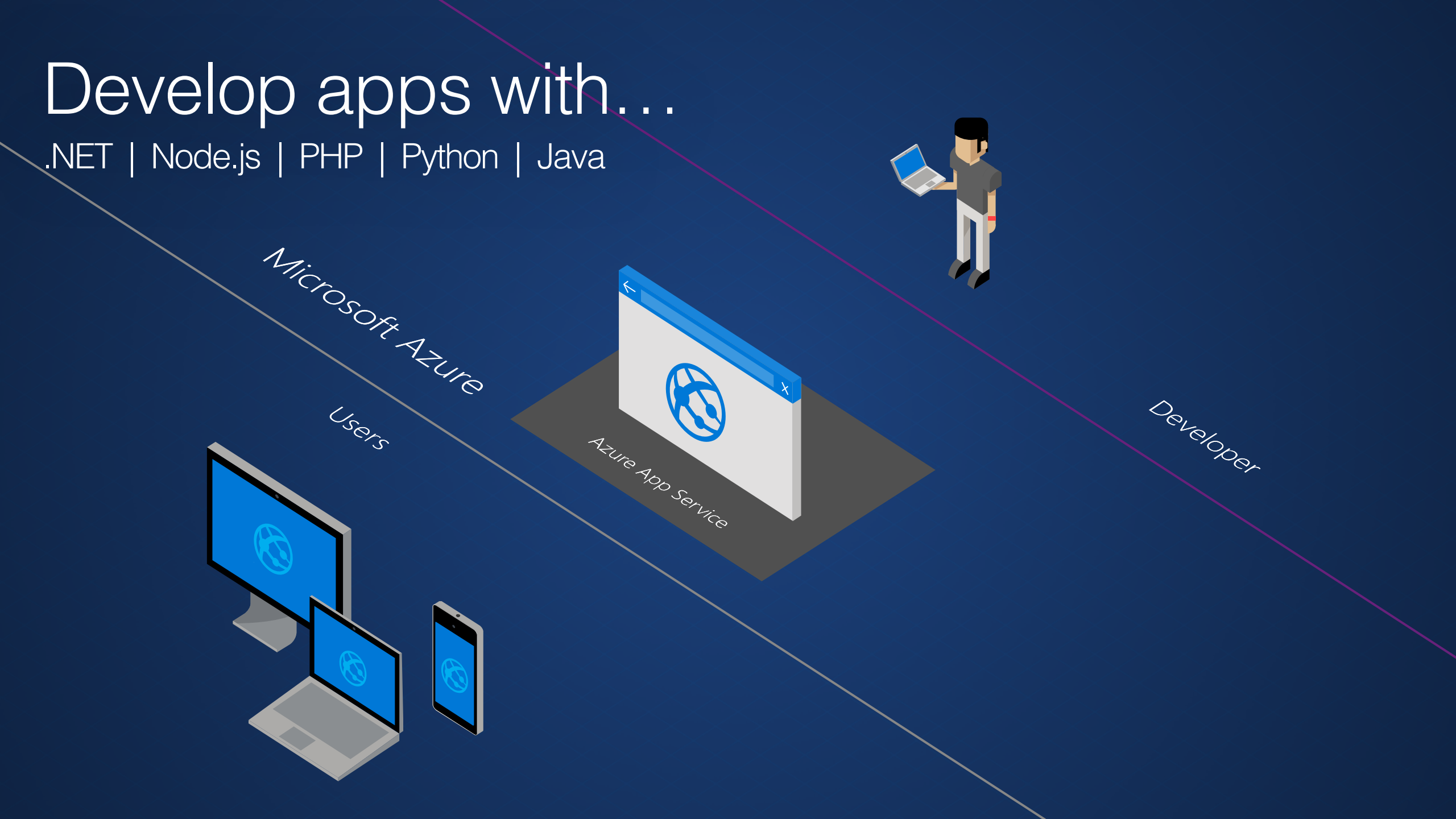
Azure App Service

Build and scale cloud apps



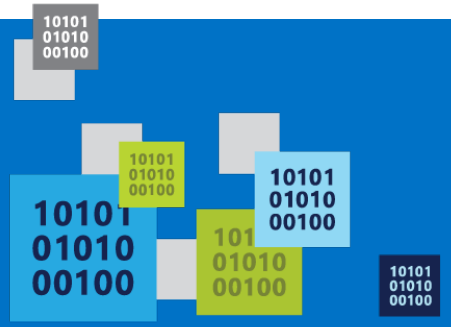
Develop apps with…

.NET | Node.js | PHP | Python | Java

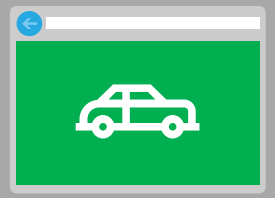


Benefits of App Services

- Automatic OS patching
- Enterprise-grade security
- High availability
  - Automated scale out/in
  - Built-in load balancing
- Supports many languages and platforms
  - .NET, Node.js, Python, Ruby and many more
- Easy continuous deployment
  - Continuous delivery from third-party source control providers
  - Built-in Git repo

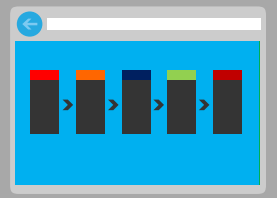


Web Apps



Customer site

Logic Apps



Order Completed

Mobile Apps

Administration App

Customer App

API Apps



WEB APPS

Web apps run as-is
no changes required

Full capability set available including:

- .NET, Node.js, Java, PHP, and Python
- WebJobs for long running tasks
- Integrated VS publish, remote debug…
- CI with GitHub, BitBucket, VSO
- Auto-load balance, Autoscale, Geo DR
- Virtual networking and hybrid connections
- Site slots for staged deployments

WebJobs: Light-weight CPU Intensive Tasks





run.cmd, run.batrun.exe run.ps1 run.sh run.php run.py run.js

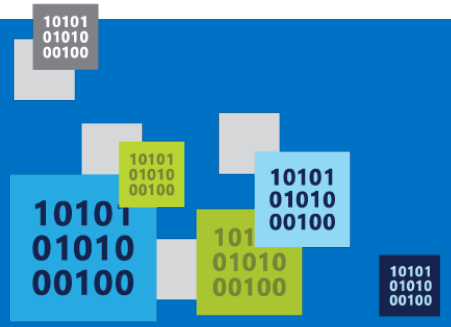
Job Type: On-Demand, Scheduled, Continuous

WebJobs SDK Feature:

Scale: Singleton, Multi-instance

BlobTrigger, TableTrigger, QueueTrigger, ServicebusTrigger

Deployment: Portal, Visual Studio, CLI, Git



API APPS

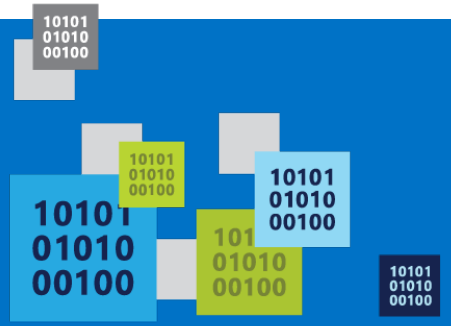
Create, consume and host APIs more easily

Easily use cloud or custom APIs:

- Dozens of built-in APIs for popular SaaS
- An ecosystem of APIs for any need
- Create and publish custom, reusable APIs
- Visual Studio tooling with one click publish and remote debugging
- Automatic client SDK generation for many languages

Benefits of API Apps

- Bring your API as-is
  - .NET Web API
  - Node.js + Express
  - Java
  - PHP
  - Many other technologies
- Connect easily to SaaS platforms



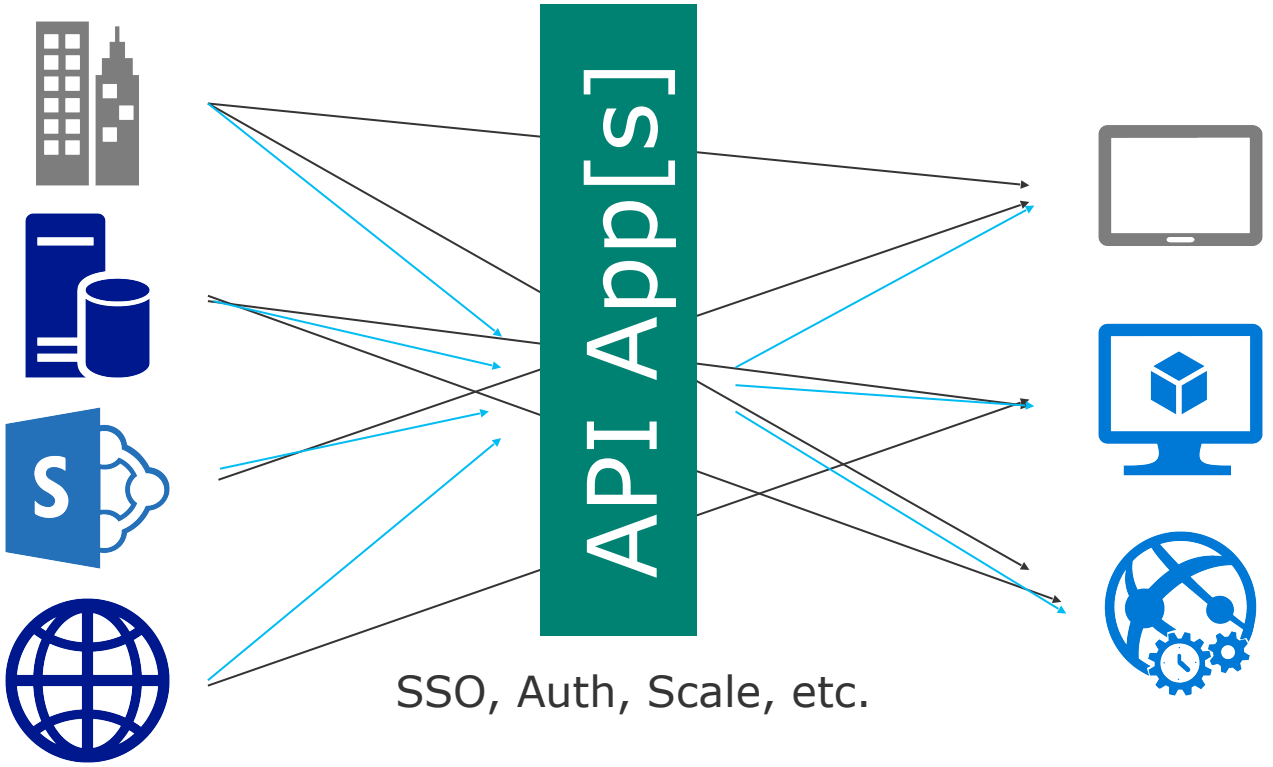
Managed Middle Tier

On Premise Applications

Databases in Azure VMs

SharePoint Online

Cloud-Hosted Web Apps



LOGIC APPS

Automate SaaS and
on-premises systems

New Logic Apps for easy automation

- No code designer for rapid creation
- Dozens of pre-built templates to get started
- Out of box support for popular SaaS and on-premises apps
- Use with custom API apps of your own
- Biztalk APIs for expert integration scenarios

Built-in API Connectors

Connectors

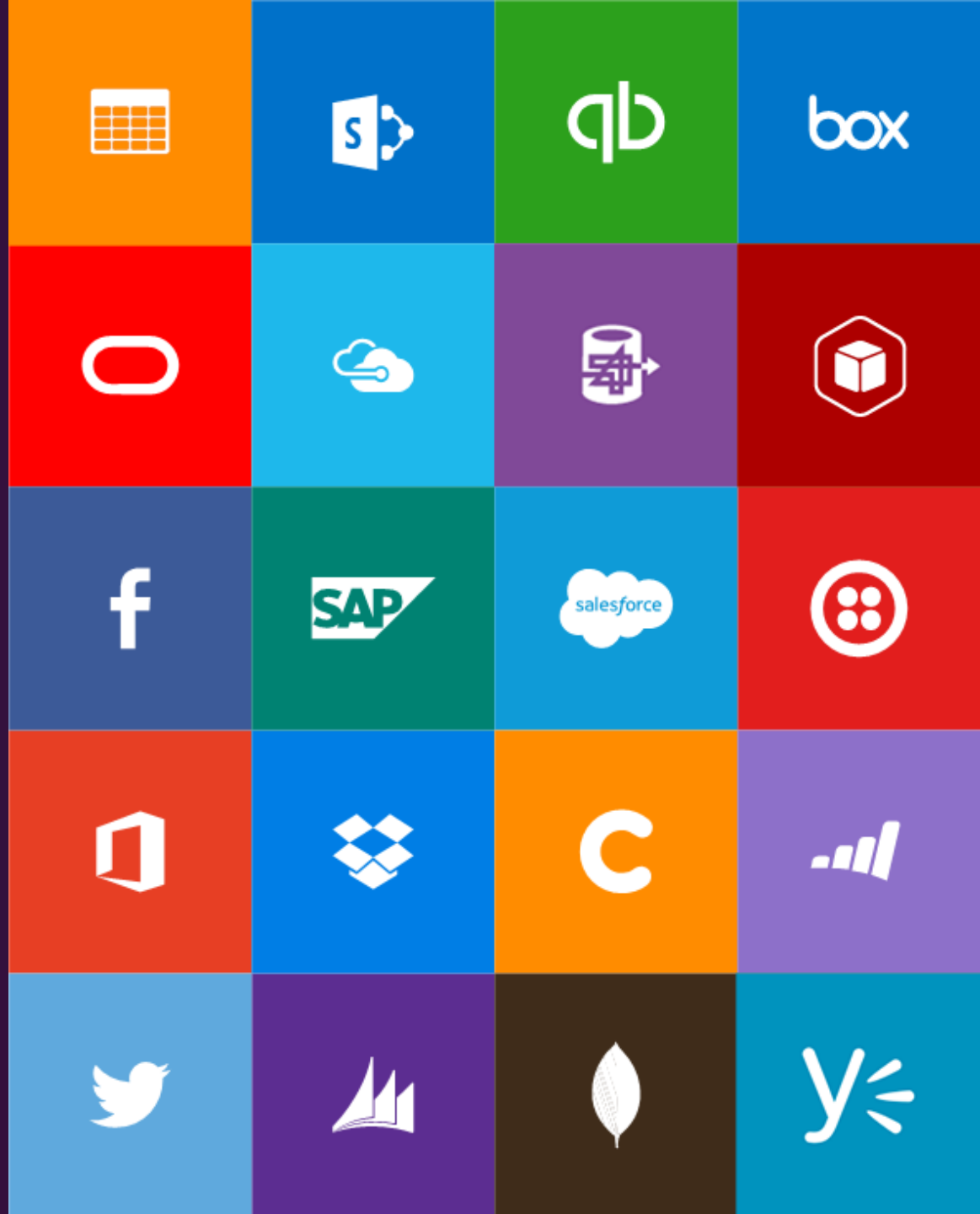
- Box
- Chatter
- Delay
- Dropbox
- Azure HD Insight
- Marketo
- Azure Media Services
- OneDrive
- SharePoint
- SQL Server
- Office 365
- Oracle
- QuickBooks
- SalesForce
- Sugar CRM
- SAP
- Azure Service Bus
- Azure Storage
- Timer / Recurrence
- Twilio
- Twitter
- IBM DB2
- Informix
- Websphere MQ
- Azure Web Jobs
- Yammer
- Dynamics CRM
- Dynamics AX
- Hybrid Connectivity

Protocols

- HTTP, HTTPS
- File
- Flat File
- FTP, SFTP
- POP3/IMAP
- SMTP
- SOAP + WCF

BizTalk Services

- Batching / Debatching
- Validate
- Extract (XPath)
- Transform (+Mapper)
- Convert (XML-JSON)
- Convert (XML-FF)
- X12
- EDIFACT
- AS2
- TPMOM
- Rules Engine



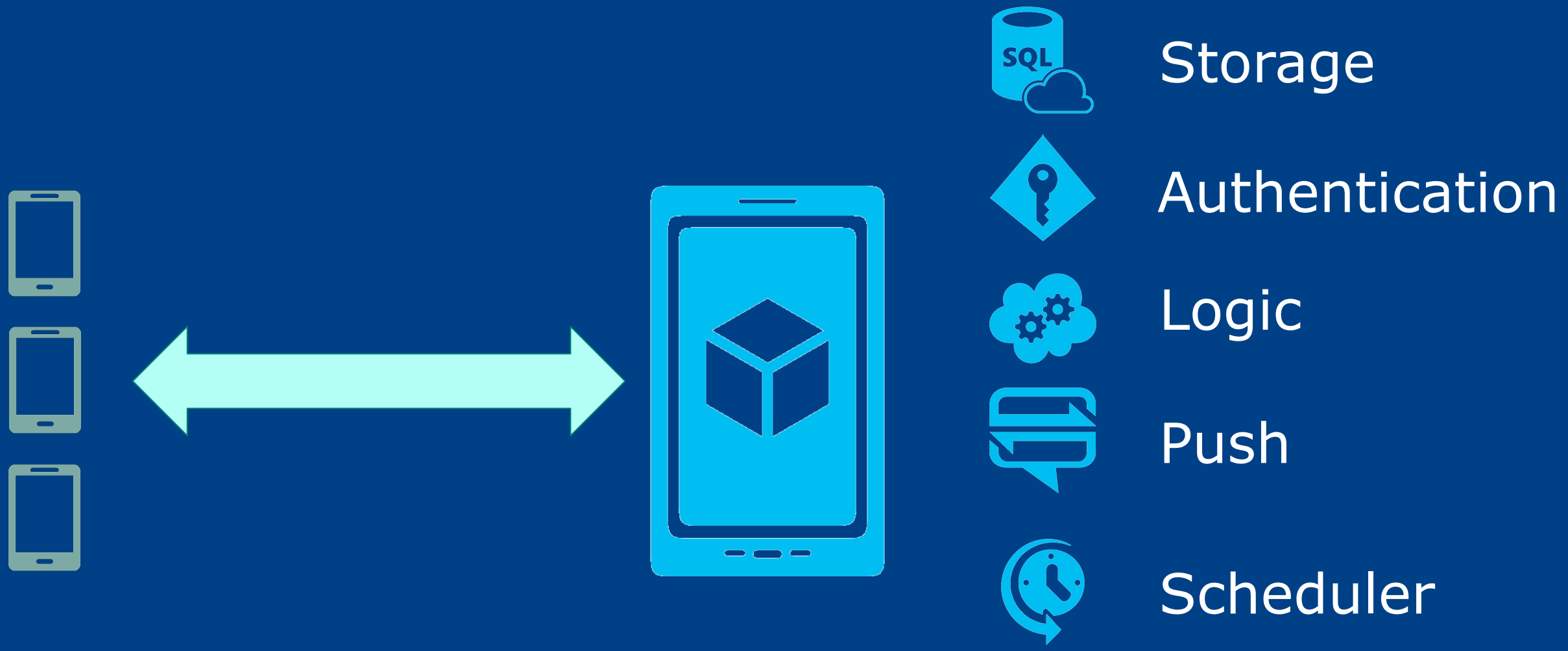
MOBILE APPS

Mobile services plus
a whole lot more

New capabilities for Mobile apps:

- Webjobs for long running tasks
- CI with GitHub, BitBucket, VSO
- Auto-load balance, Autoscale, Geo DR
- Virtual networking and hybrid connections
- Site slots for staged deployments

Azure Mobile Services



Structured Storage

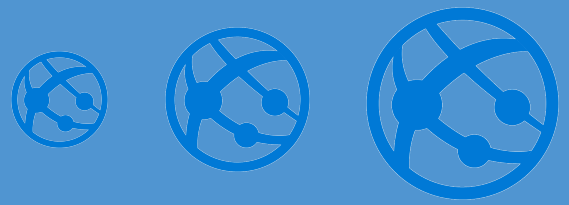
- Powered by SQL Database
- Supports rich querying capabilities
- Dynamic Schematization
- Data management in:
  - Azure Portal
  - SQL Portal (Silverlight)
  - SQL Management Studio
  - REST API
  - Azure CLI Tools
  - SQL CLI

Custom API

- Non-table based endpoints
- Accessible from
  - GET
  - POST
  - PUT
  - PATCH
  - DELETE
- For node.js logic in scripts like table endpoints
- For .NET delivered through a WebAPI
- Expose any functionality you want

Scaling Up vs. Scaling Out

Scale Up



Vary the VM size

1 Core w/ 1.75 GB RAM
2 Cores w/ 3.5 GB RAM
4 Cores w/ 7 GB RAM

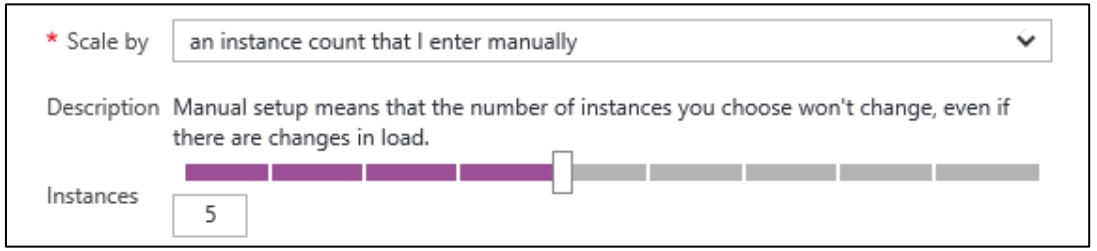
Scale Out

Vary the VM count

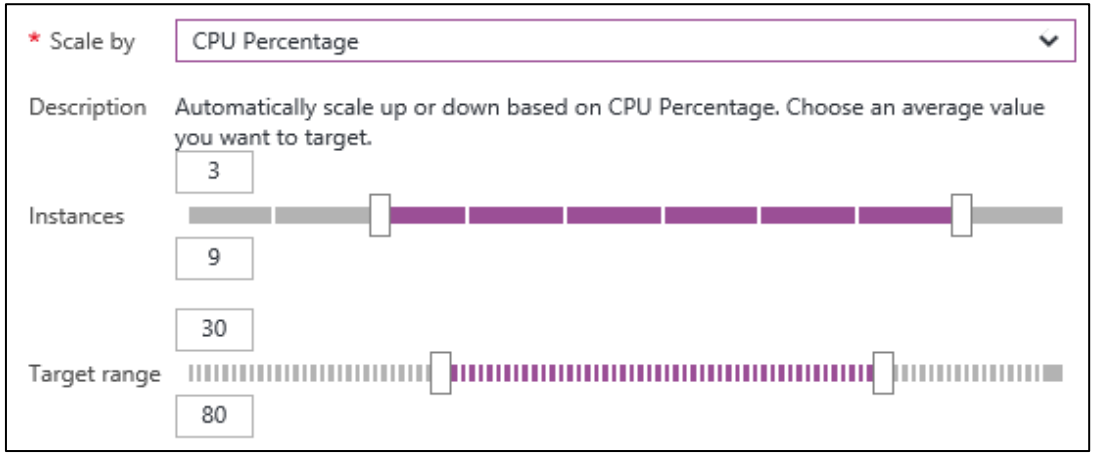
Max 3 instances*
Max 10 instances
*Max 20/50** instances*

Manual Scaling vs. Auto-Scaling

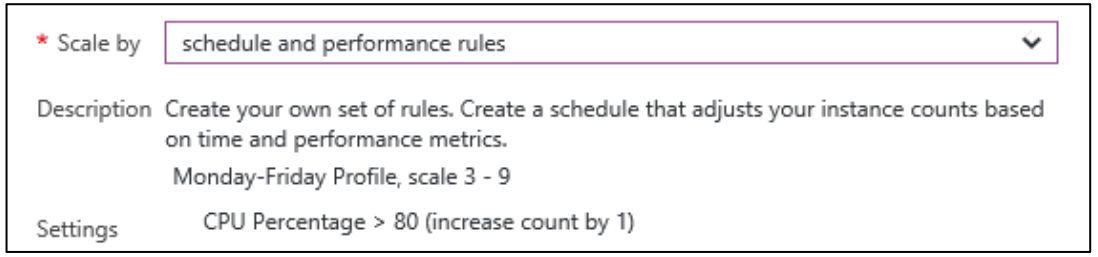
Manual – Scale via portal or scripts



Auto – CPU Percentage



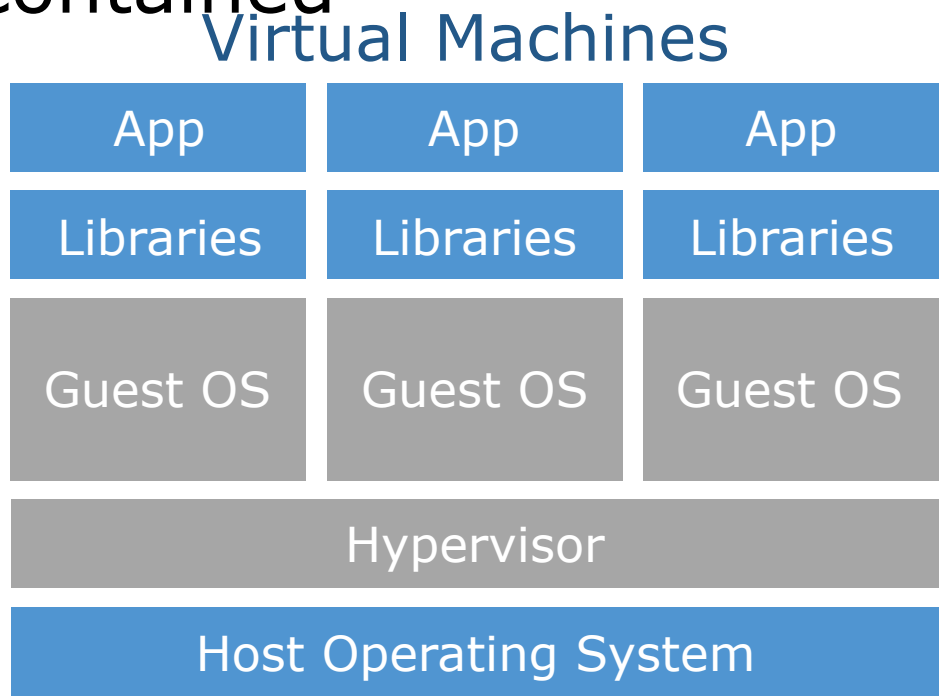
Auto – Schedule & Performance Rules

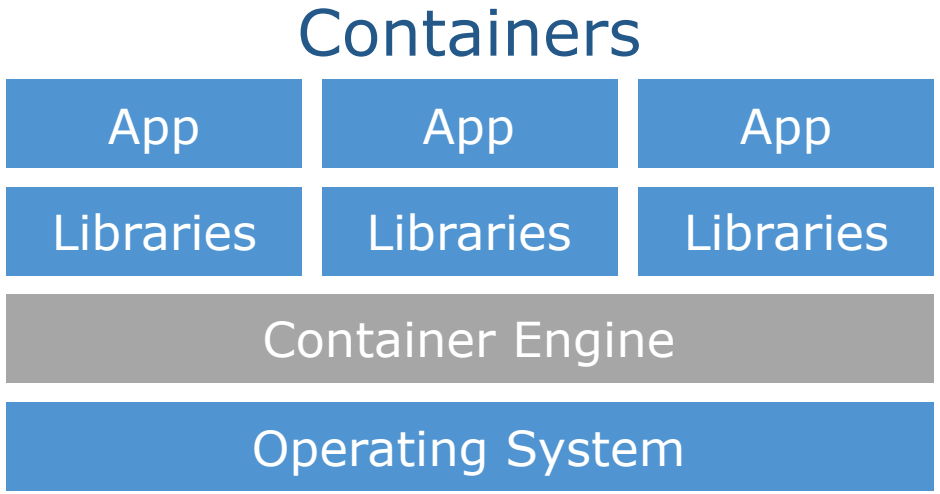


Azure Container Service

Containers

- Lightweight alternative to virtual machines
- Smaller, less expensive, faster to start up, and self-contained



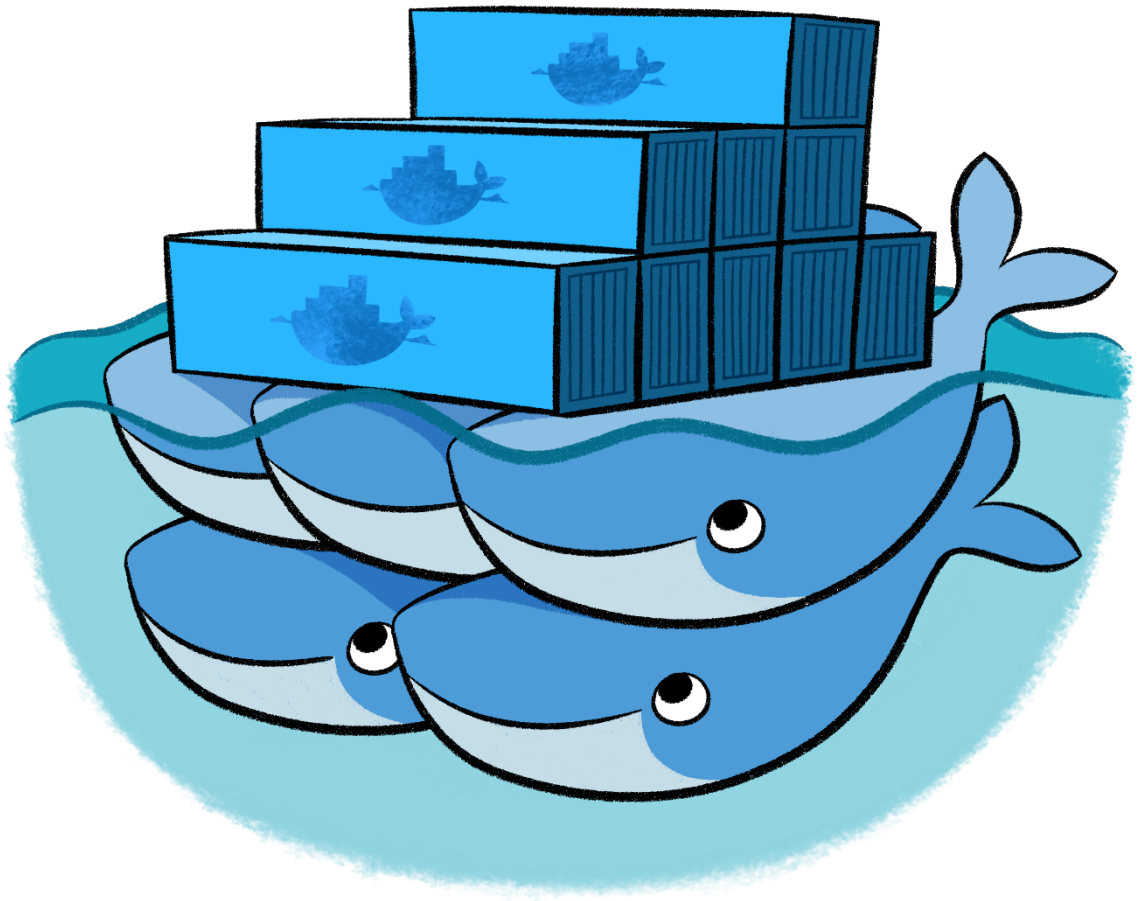


Docker

- Leading open-source containerization platform

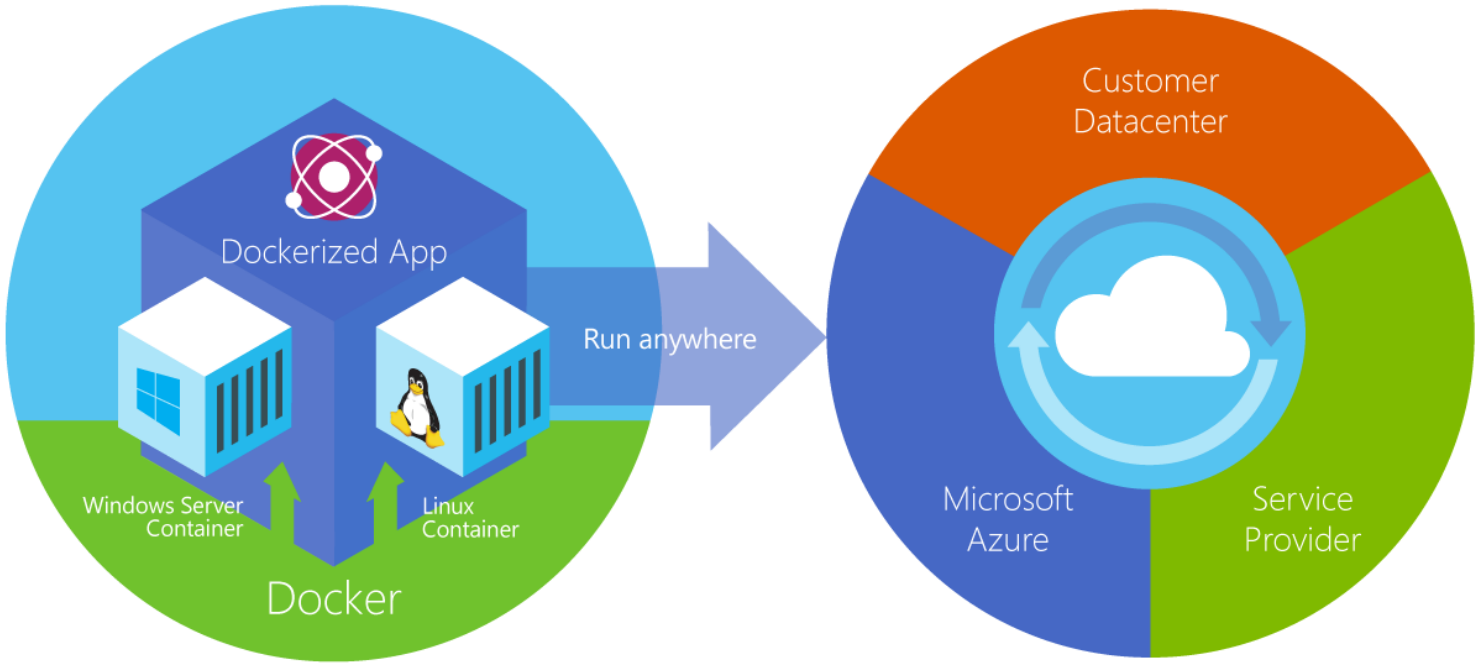
  Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in

- Supported natively in Azure via Azure Container Service

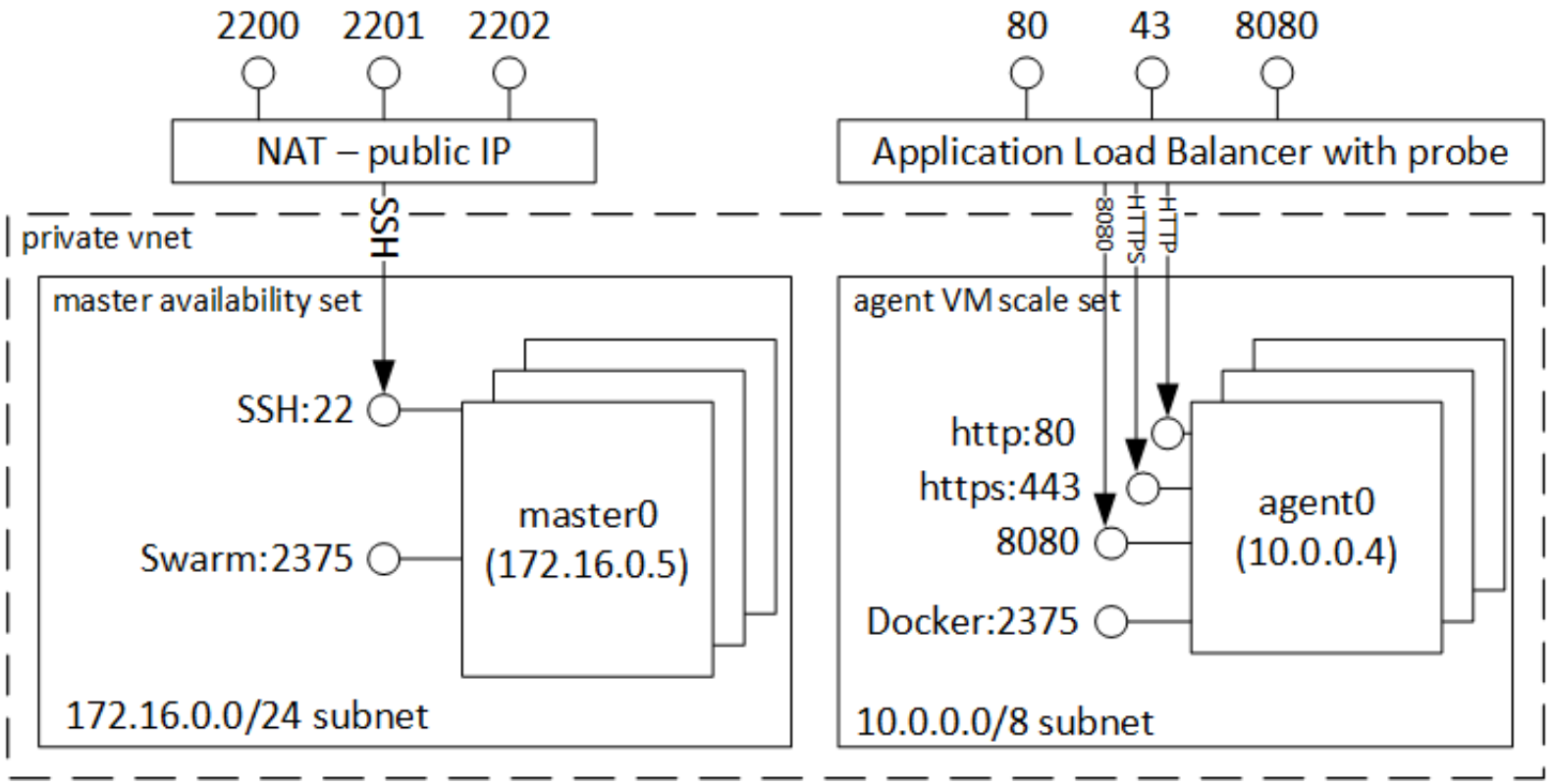


Azure Container Service

- Provides robust, ready-to-use Docker hosting environment
- Uses open-source orchestration tools (DC/OS and Swarm)

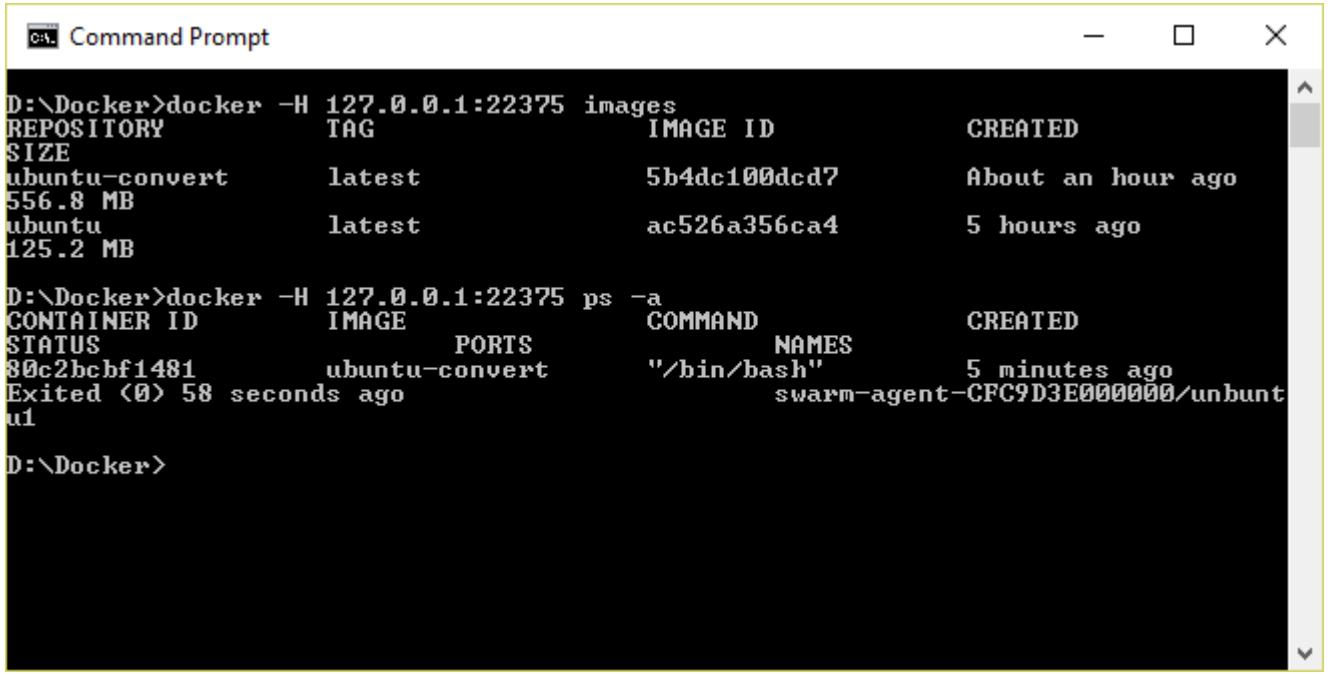


Clustering with Docker Swarm



Docker Client

- Command-line interface for Docker, available for Linux, OS X, and Windows (available separately or as part of Docker Toolbox)



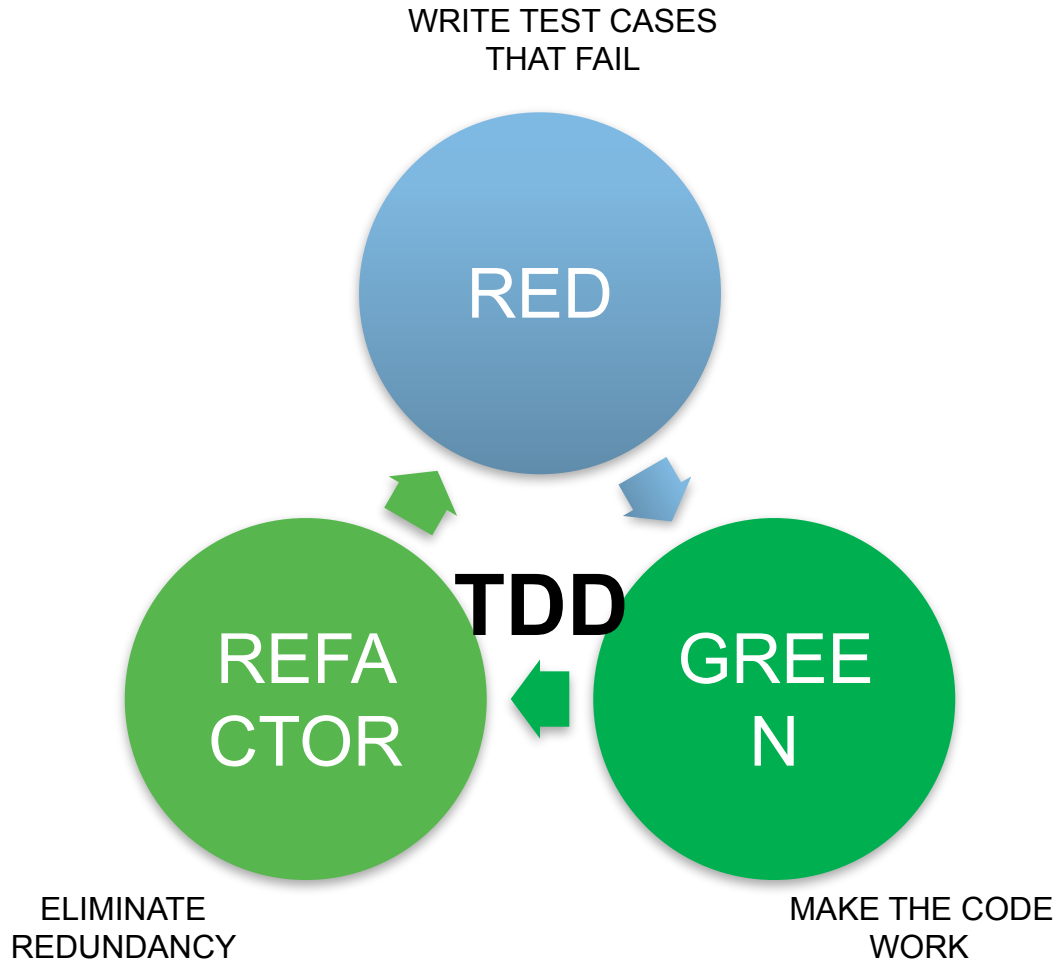
DevOps

Agile

- Tip to tail: Dev to Customer
  - Pain points
- The three pillars of the Agile stool:
  - Automation
  - Testing
  - Skill
- DevOps fits like a glove

Dev/Test Cycle

- Zealots of automation
- Time is short—getting to market needs a well-paved path
- Nothing manual
- Especially Q/A



DevOps Practices

The Bottlenecks in the Flow

- Environment creation
- Code deployment
- Test setup and run (menti)
- Overly tight architecture
- Development
- Product management

“In November 2011, running even the most minimal test for CloudFoundry required deploying to 45 virtual machines, which took a half hour. This was way too long, and also prevented developers from testing on their own workstations.

By using containers, within months, we got it down to 18 virtual machines so that any developer can deploy the entire system to single VM in six minutes.”

Elisabeth Hendrickson, Director of Quality Engineering, Pivotal Labs

Google Dev and Ops (2013)

- 15,000 engineers, working on 4,000+ projects
- All code is checked into one source tree (billions of files!)
- 5,500 code commits/day
- 75 million test cases are run daily

“Automated tests transform fear into boredom.”

Eran Messeri, Google

Inject Failures Often

The Netflix Tech Blog

We've sometimes referred to the Netflix software architecture in AWS as our Rambo Architecture. Each system has to be able to succeed, no matter what, even all on its own. We're designing each distributed system to expect and tolerate failure from other systems on which it depends.

One of the first systems our engineers built in AWS is called the Chaos Monkey. The Chaos Monkey's job is to randomly kill instances and services within our architecture. If we aren't constantly testing our ability to succeed despite failure, then it isn't likely to work when it matters most – in the event of an unexpected outage.

NETFLIX

You Don’t Choose Chaos Monkey—Chaos Monkey Chooses You

The 2014 AWS Reboot

“When we got the news about the emergency EC2 reboots, our jaws dropped. When we got the list of how many Cassandra nodes would be affected, I felt ill. Then I remembered all the Chaos Monkey exercises we’ve gone through. My reaction was, ’Bring it on!’.”

Christos Kalantzis, Netflix Cloud DB Engineering

Add Ops into Dev

Enhance Service Design with Operational Knowledge:

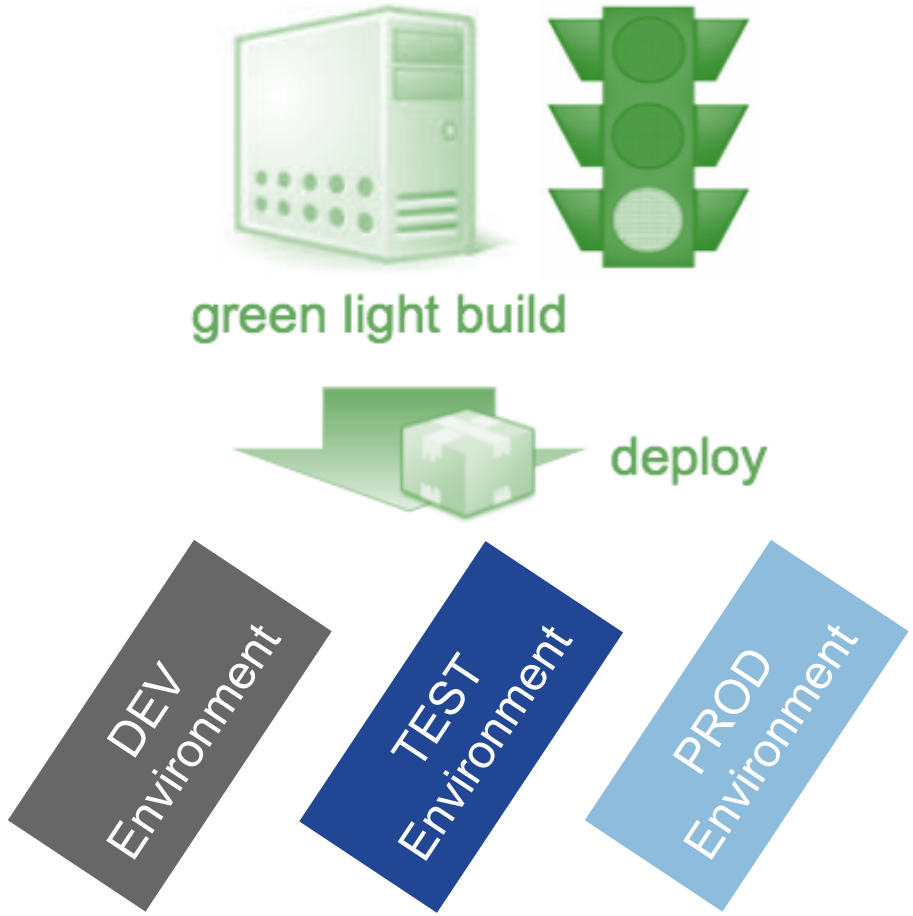
- Reliability
- Performance
- Security
- Test them

Build feedback paths back from Production:

- Foster a culture of responsibility
  - Whether your code passes test, gets deployed, and stays up for users is your responsibility – not someone else
- Make Development better with Ops:
  - Production-like environments
  - Power tooling

Continuous Deployment

- Extension of continuous integration
- First: automate deployment
- Application always known to be in a ”deployable” state



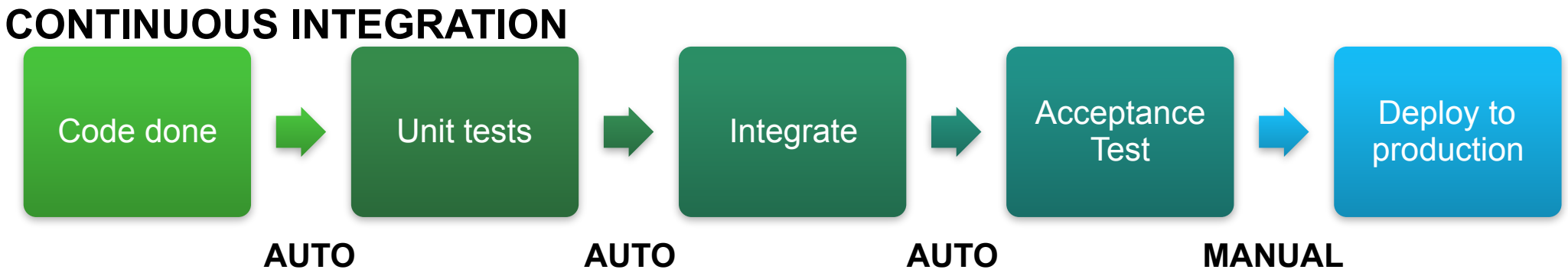
DevOps Lifecycles

Principles of Continuous Integration

- Code repository
- Automate the build
- Make the build self-testing
- Everyone commits to the main-line every day
- Every commit is built
- Build must be fast
- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automated deployments

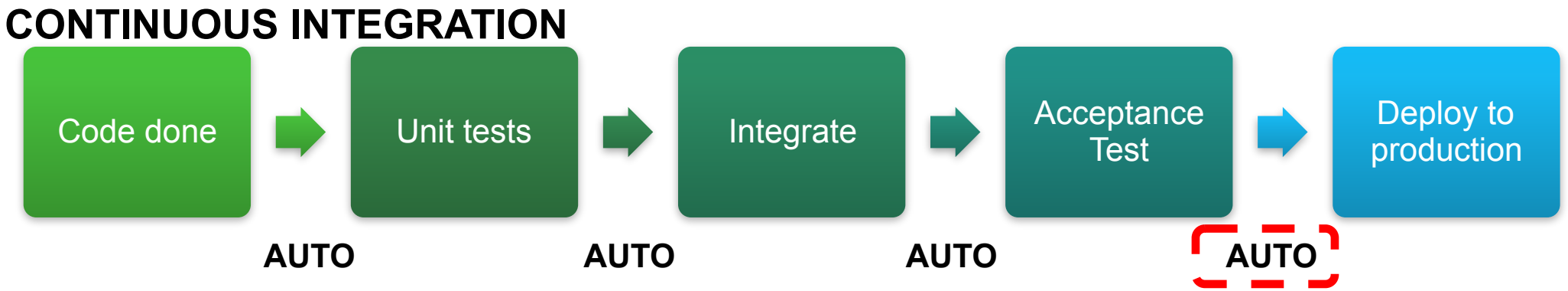
How CI Improves Efficiency

- Rapid feedback
- Reduce technical debt
- Visibility
- Builds automated
- Precursor to continuous delivery and deployment



Keys to Continuous Delivery

- Minimize shock
- Avoid off-hour, high risk, expensive deployments
- Know your rollback plan
- Build in health checks



Feedback Loops

- Understanding and responding to the needs of all customers (internal and external)
- Shorten feedback loops
- Feedback = quality

Types of Tools

- Build tools
- Test tools
- General automation
- Configuration management

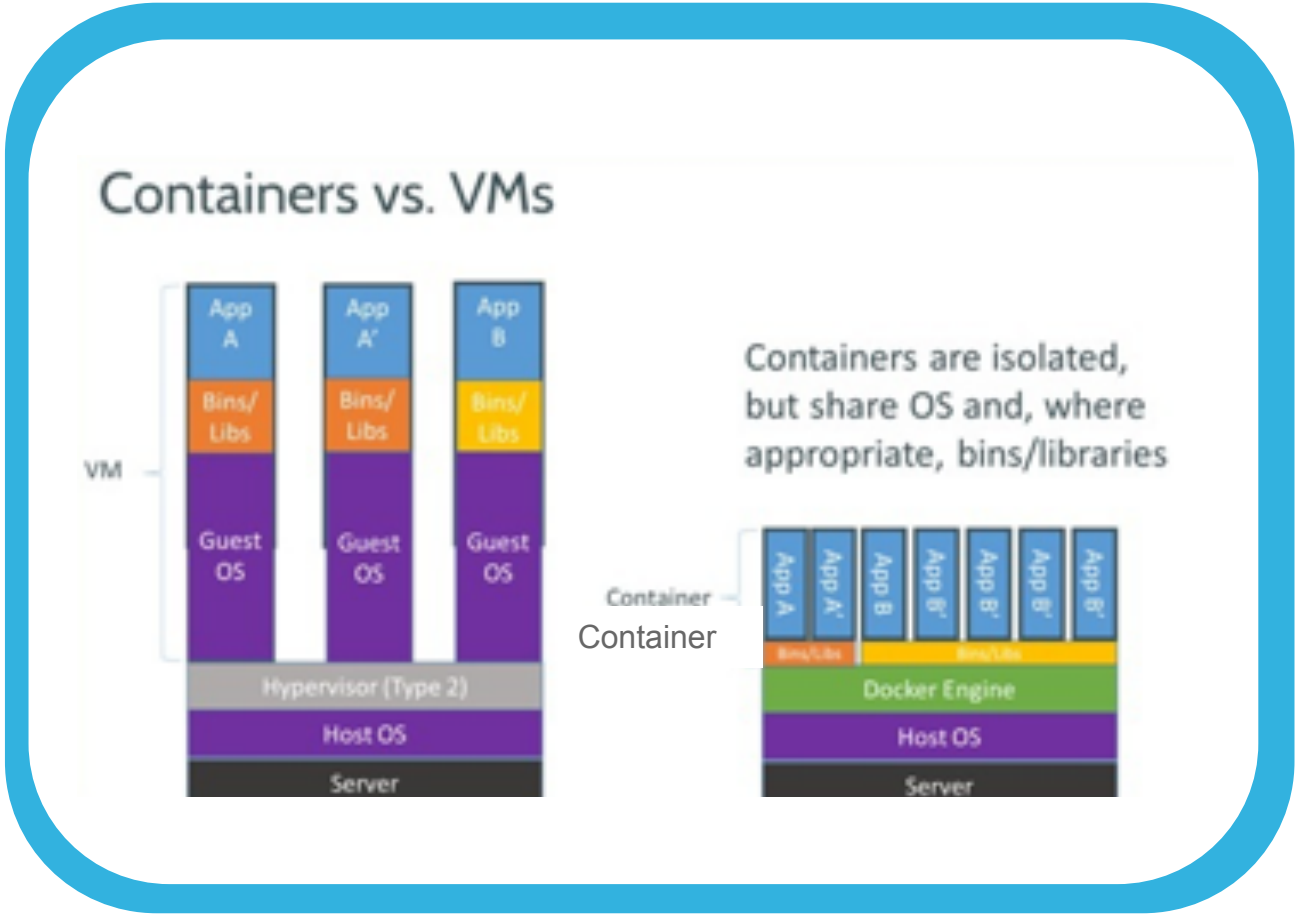
DevOps Related Technology

Virtualization

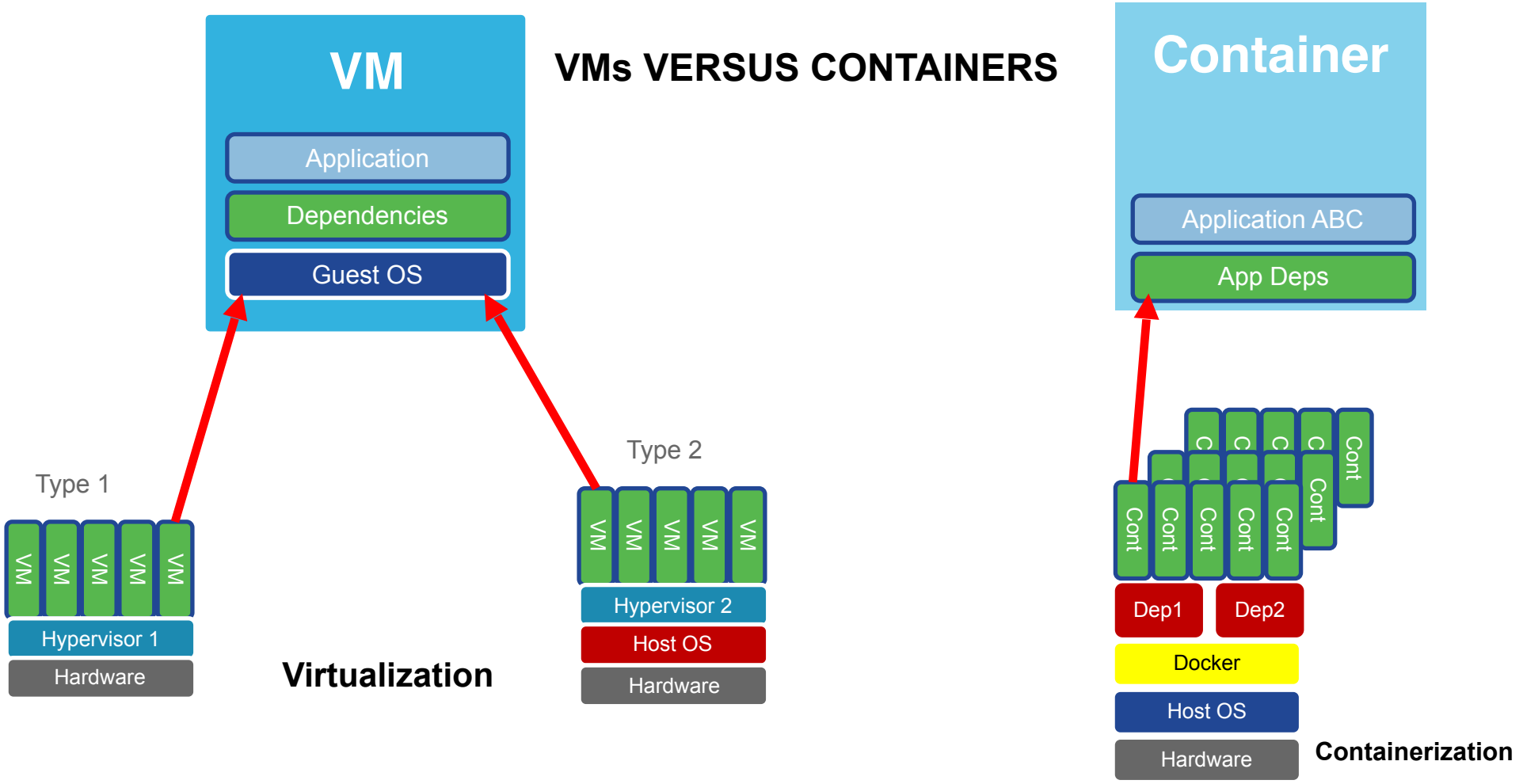
- Been around for a while
- Think VMware ESX
- Virtual Machines

Containers (Docker)

- Open source engine that automates the deployment of any application as a lightweight, portable, self-sufficient container that will run virtually anywhere.
- Based on LXC (Linux Container) and AUFS (Union FS), easy to use.
- Similar to VM as end-user with different features



Docker (What is it?)



Current Container Issues

- Networking
- Service Orchestration
- Security

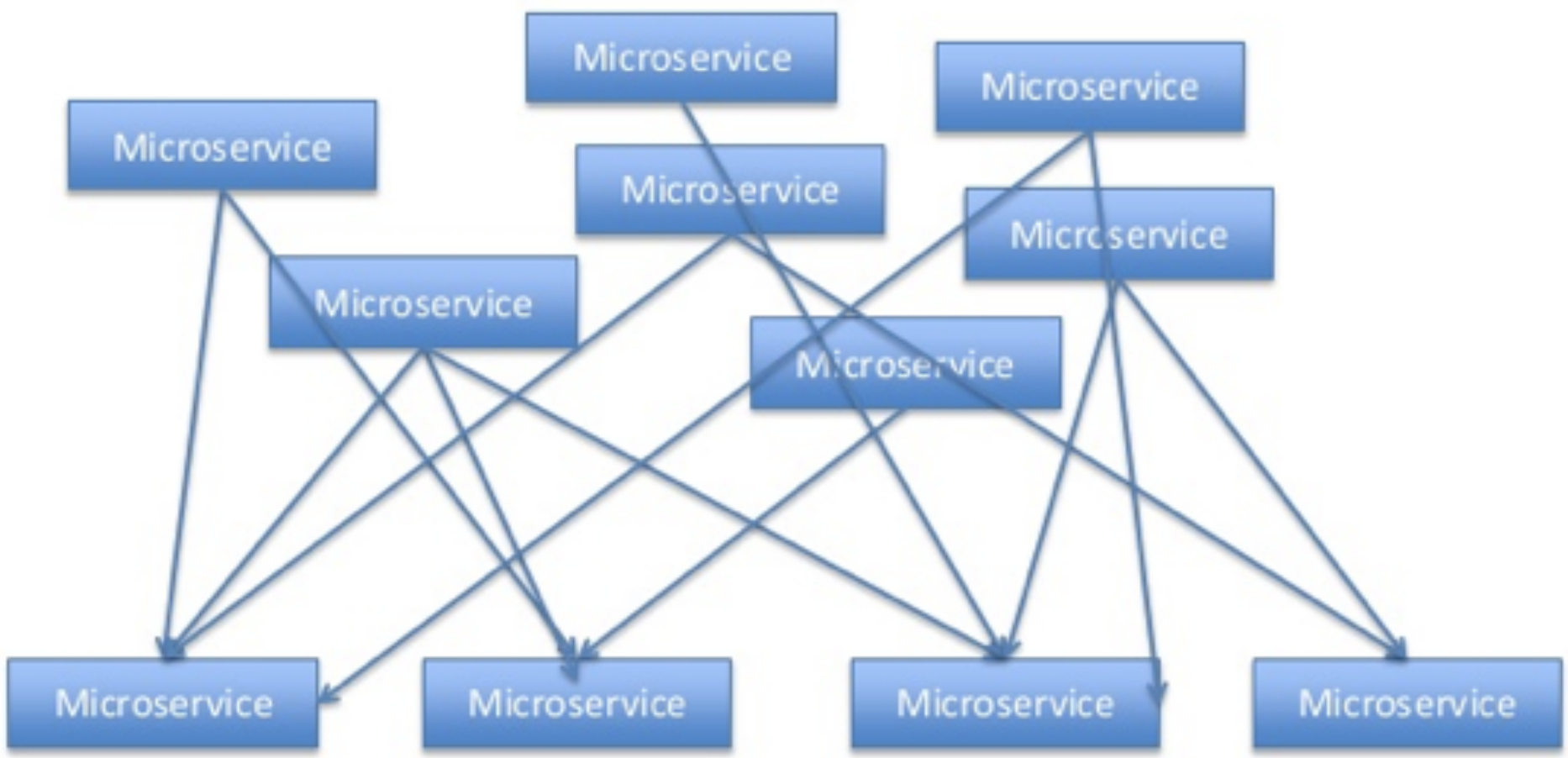
Docker Alternatives

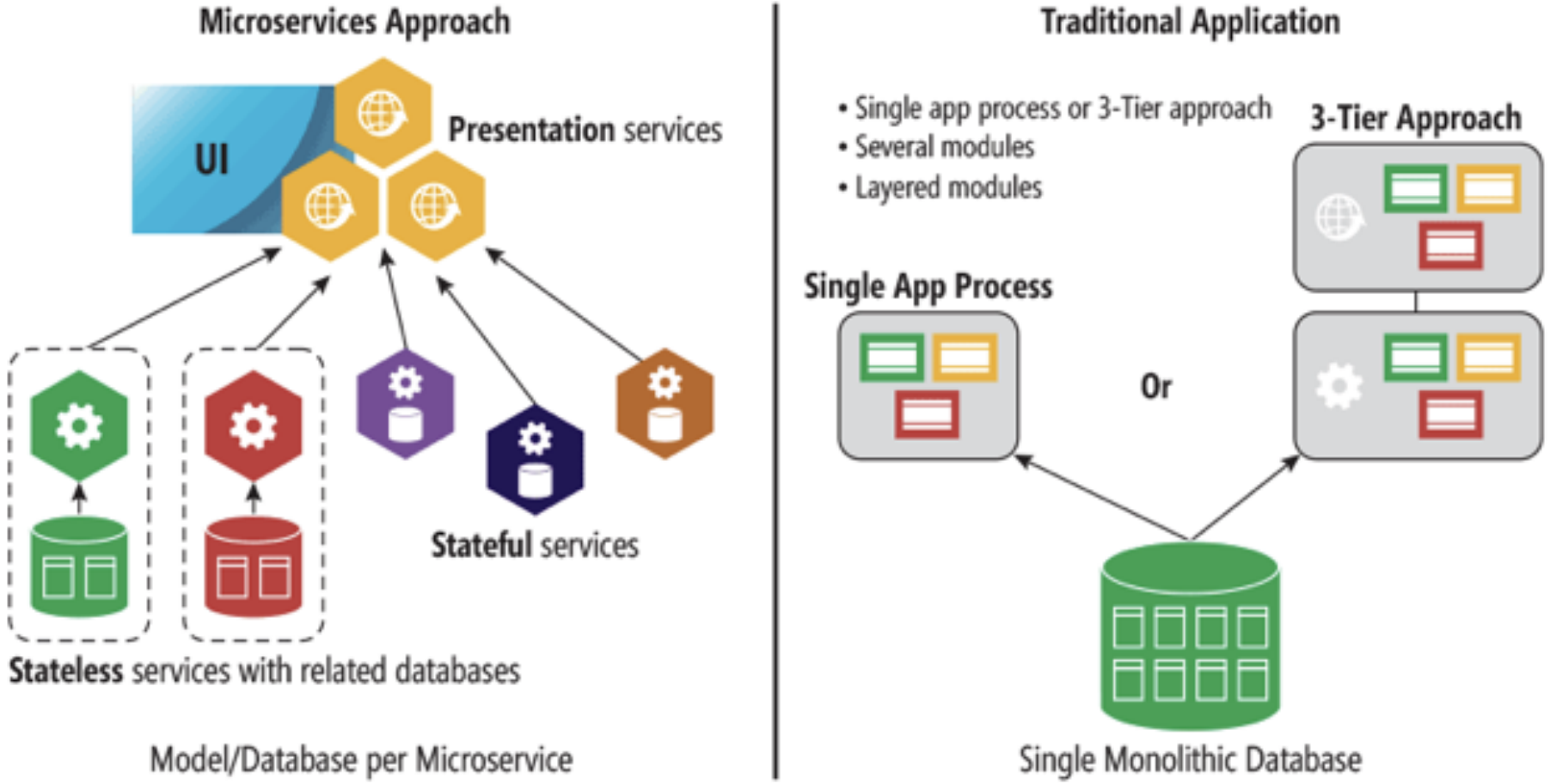
- Rocket
- Microsoft Drawbridge
- LXD (Canonical)

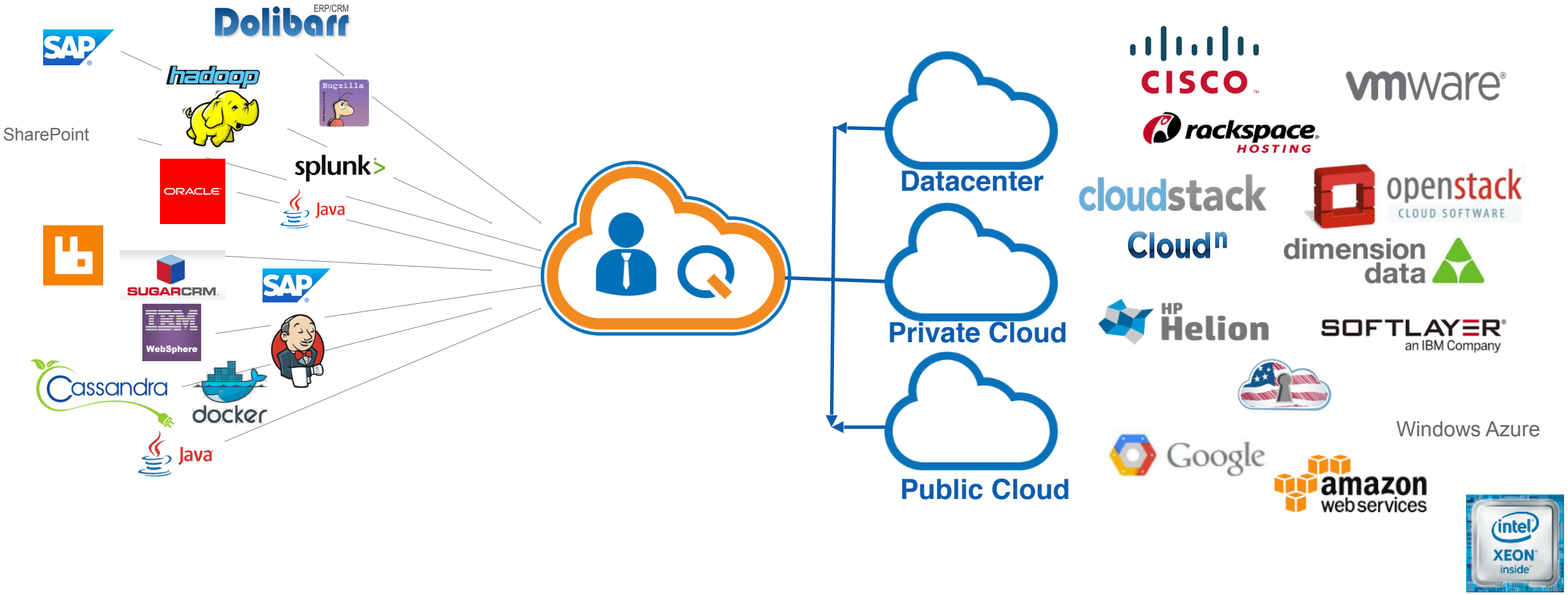
Microservices

- Architectural pattern based on a stateless service model
- Granular services targeted at providing a discrete piece of application functionality
- Couples with the container model promoted by Docker
- Services are elastic and provisioned on demand in a container as needed

Distributed Service Layer Microservices Architecture




Microservices Approach
UI
Presentation services
Stateful services
Stateless services with related databases
Model/Database per Microservice
Traditional Application
• Single app process or 3-Tier approach
• Several modules
• Layered modules
3-Tier Approach
Single App Process
Or
Single Monolithic Database


CliQr
ERP/CRM
Dolibarr
SAP
hadoop
Bugzilla
SharePoint
splunk>
ORACLE
Java
SUGARCRM
SAP
IBM
WebSphere
Cassandra
docker
Java
Datacenter
Private Cloud
Public Cloud
CISCO
vmware
rackspace
HOSTING
cloudstack
openstack
CLOUD SOFTWARE
Cloudn
dimension data
HP Helion
SOFTLAYER
an IBM Company
Google
Windows Azure
amazon web services
intel
XEON
inside

Common Use Cases

Application Migration & Management

- New and existing applications
- Start with single application, single cloud
- Workload migration. Datacenter consolidation

DevOps and Continuous Delivery

- Self-service on demand environments
- Tool-chain automated deployment
- Hybrid cloud pipeline

Dynamic Capacity Augmentation

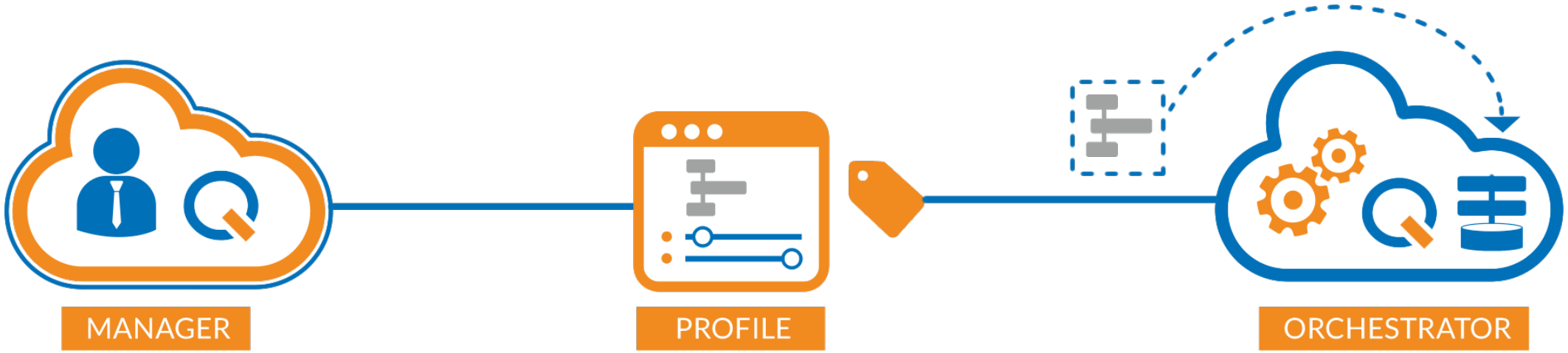
- High performance, batch and cluster
- Horizontal scaling
- Cross-environment bursting

IT as a Service with Governance

- On-demand self-service marketplace
- Single pane of glass – financial controls, usage metering, multi-tenant governance

CliQr®

CloudCenter



CloudCenter

- ✓ Single platform solution
- ✓ Unique application-defined technology
- ✓ Abstracts application from underlying cloud environment
- ✓ Ensures infrastructure adapts to meet the needs of each application

No need to:

- ✓ Write cloud specific scripting
- ✓ Write orchestration workflows
- ✓ Modify application code

CliQr®

SaaS

Software as a Service

Email
CRM
Collaborative
ERP

CONSUME

PaaS

Platform as a Service

Application
Development
Decision Support
Web
Streaming

BUILD ON IT

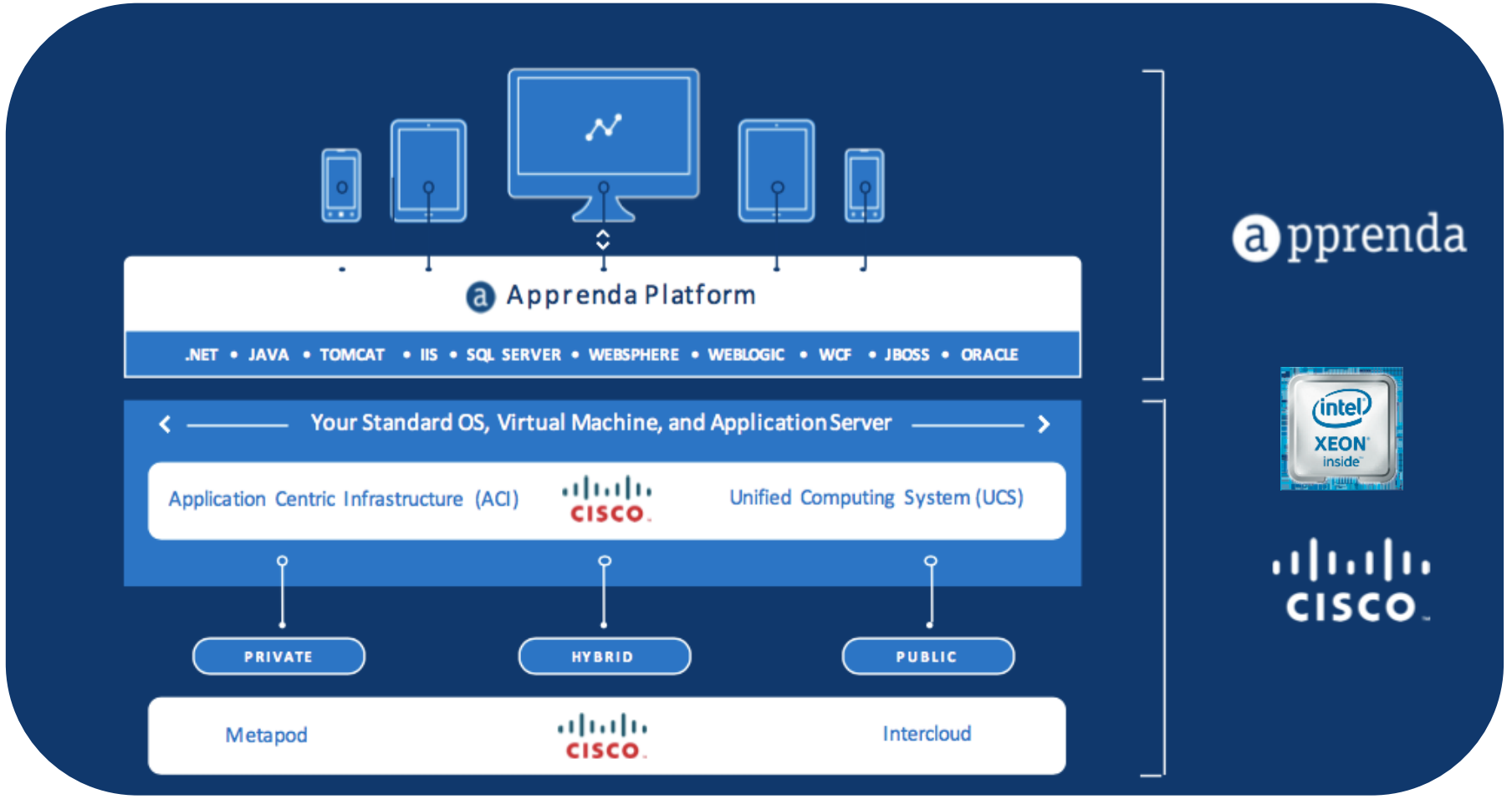
IaaS

Infrastructure as a Service

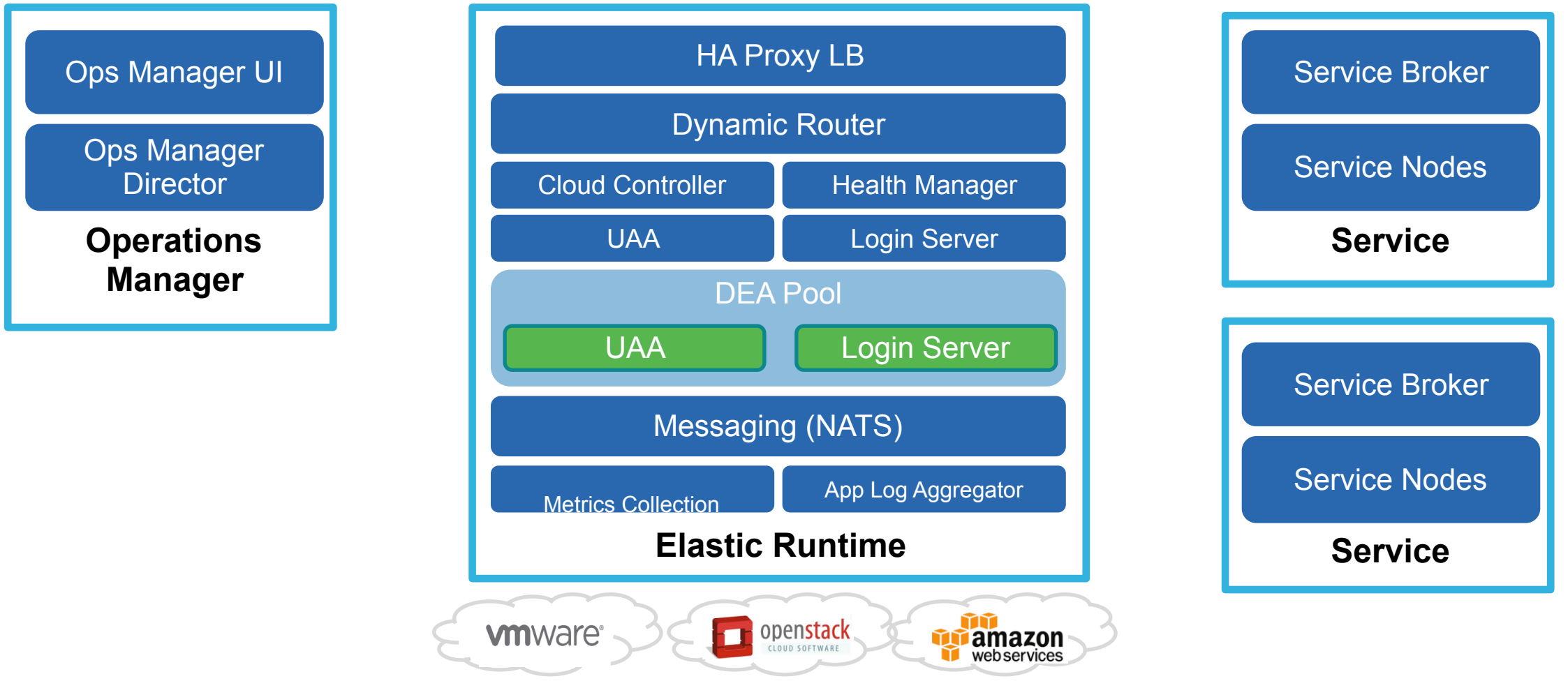
Caching
Legacy | File
Networking |
Technical
Security | SysMgmt

MIGRATE TO IT

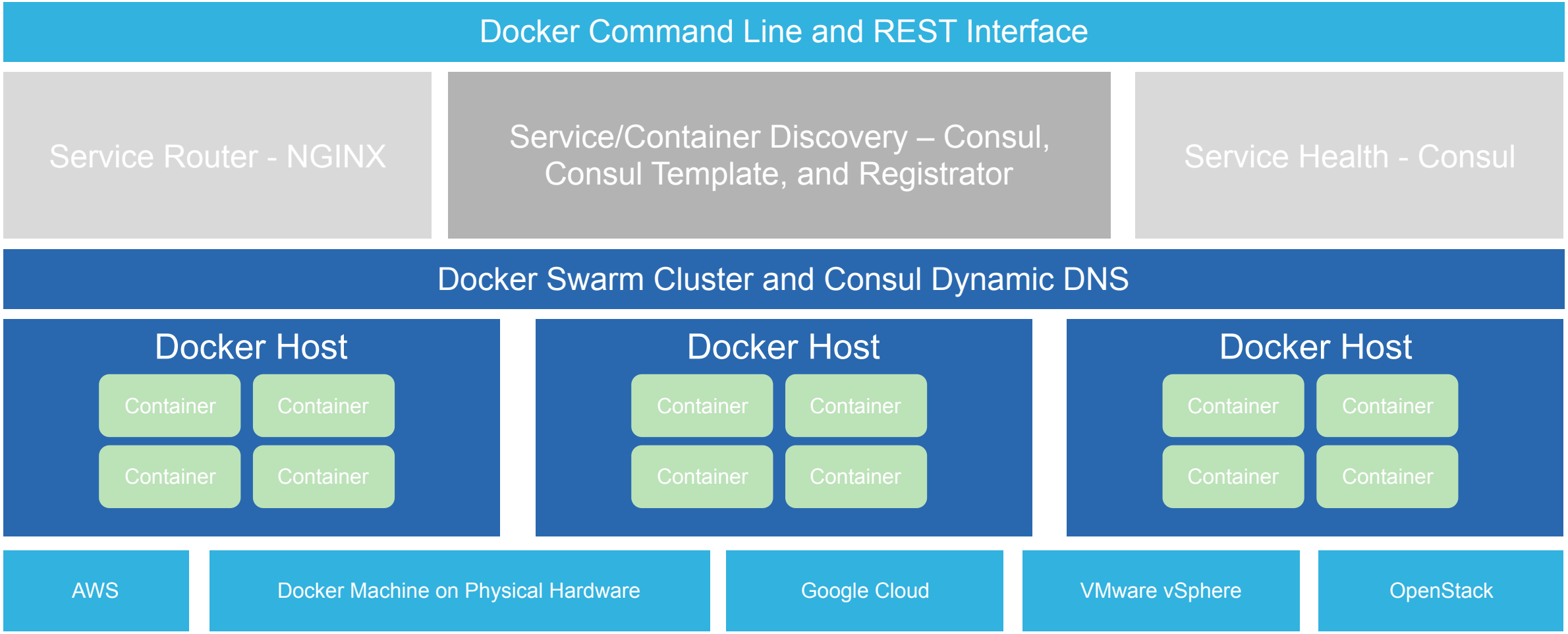
apprenda


Apprenda Platform
.NET • JAVA • TOMCAT • IIS • SQL SERVER • WEBSPHERE • WEBLOGIC • WCF • JBOSS • ORACLE
Your Standard OS, Virtual Machine, and Application Server
Application Centric Infrastructure (ACI)
CISCO
Unified Computing System (UCS)
PRIVATE
HYBRID
PUBLIC
Metapod
CISCO
Intercloud
apprenda
intel
XEON
inside
CISCO

Pivotal CF Architecture



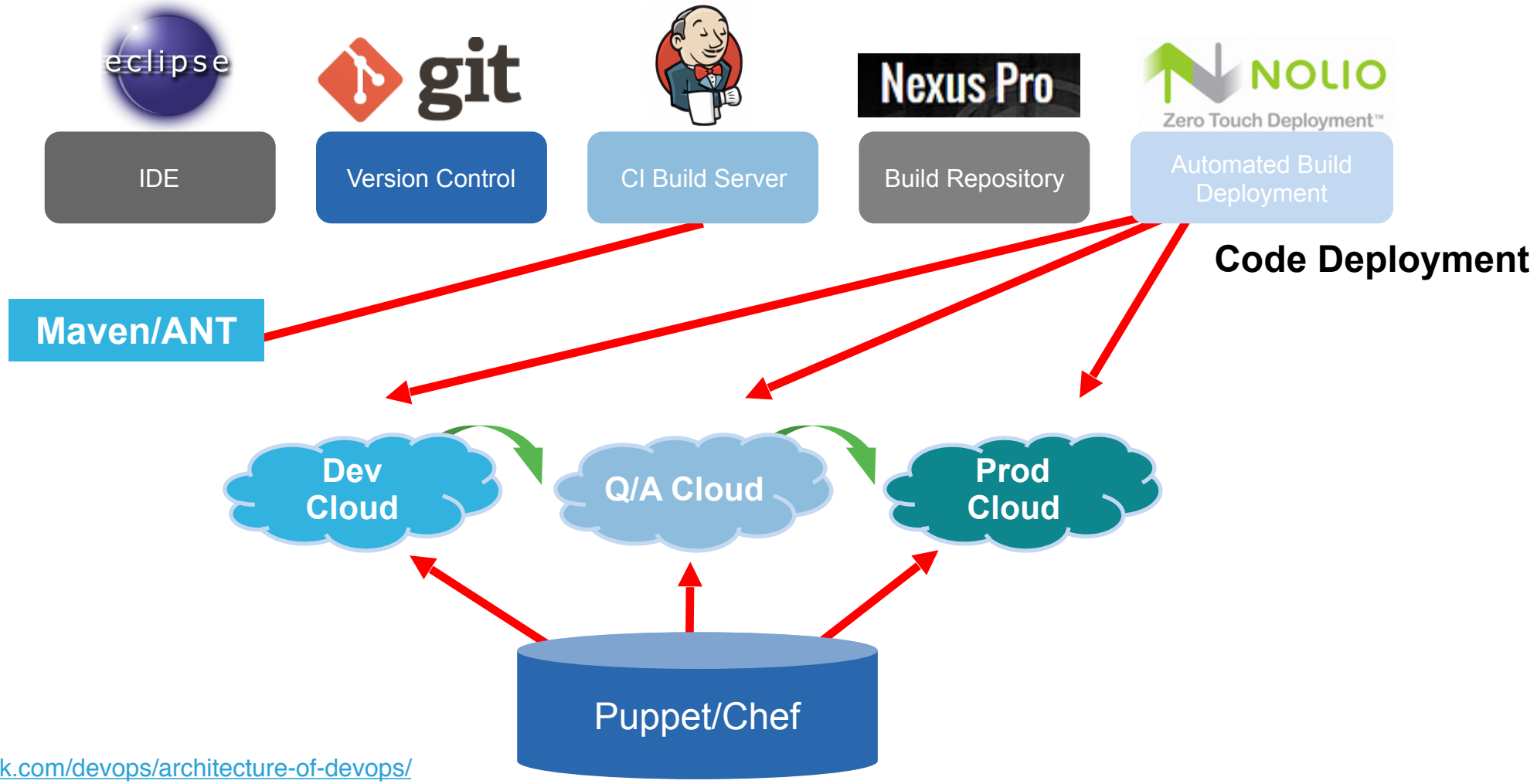
Back to Docker…



Core DevOps Technology

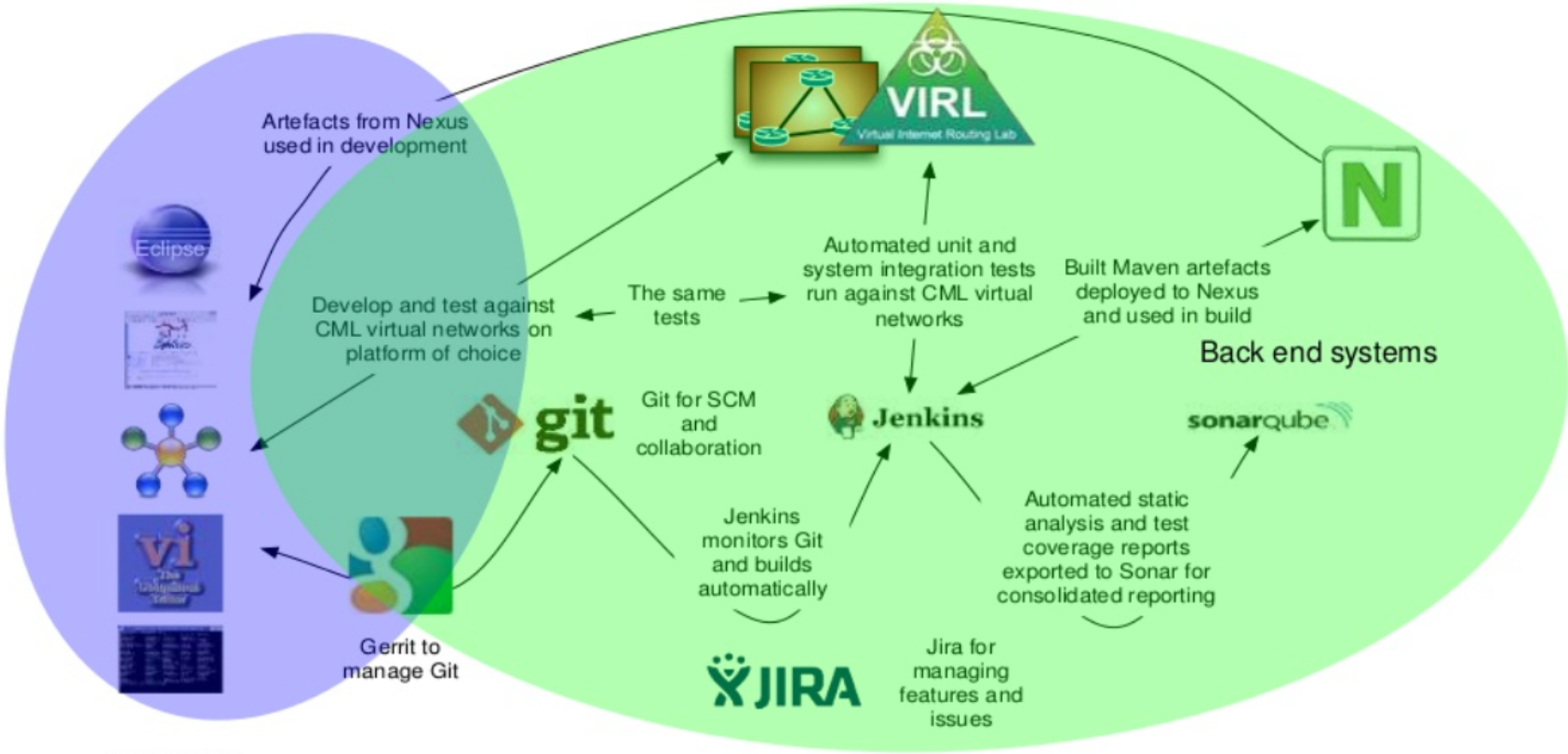
Common Elements of the Software Supply Chain

Example Reference Architecture for DevOps



SOURCE: http://agiletrick.com/devops/architecture-of-devops/

According to Cisco



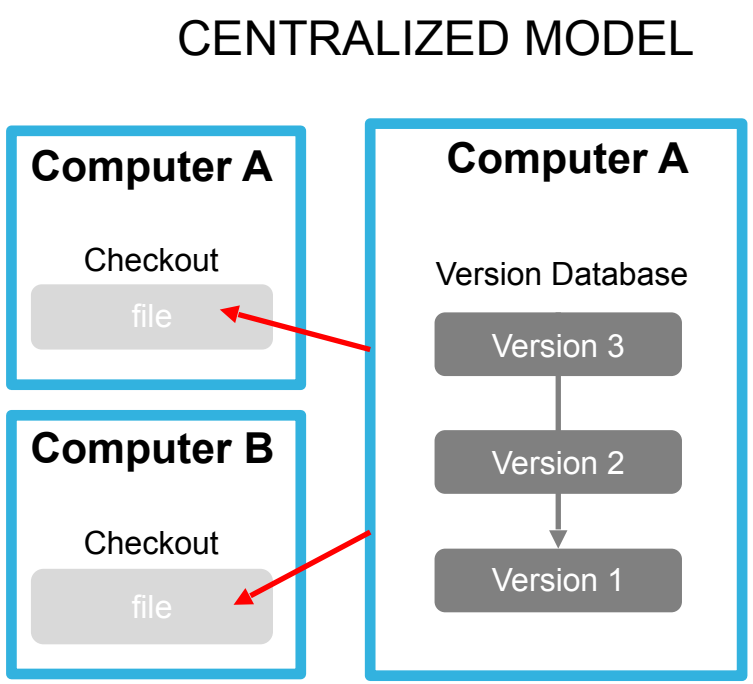
General Notes

- CAPS (Chef, Ansible, Puppet, Salt) are mainly for centrally controlling what lives inside a large number of instances. I.e. processes, files, etc.
- Terraform and CloudFormation are mainly for creating instances themselves (and other cloud resources like load balancers etc).

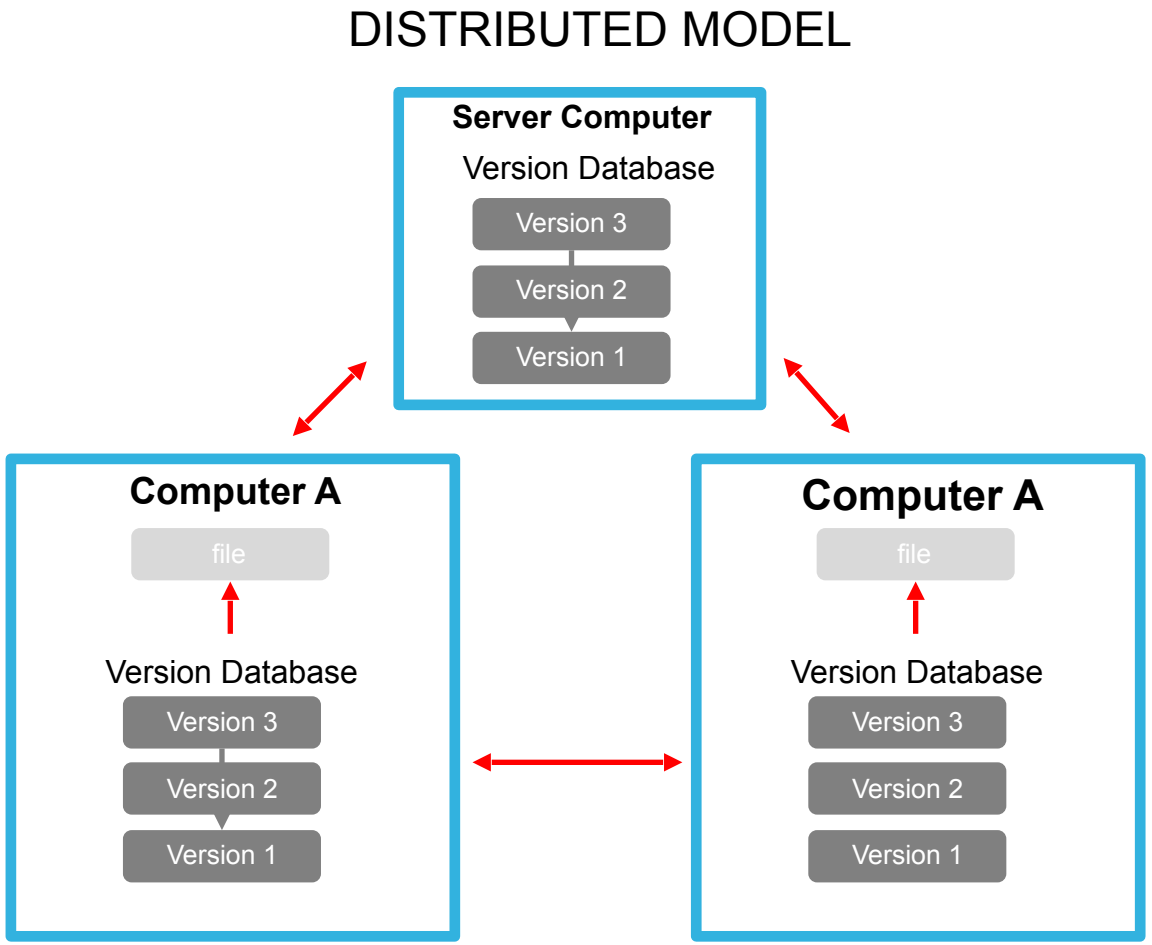
Git

- Source Control
- The backbone for open source

Git Uses a Distributed Model



(CVS, Subversion, Perforce)

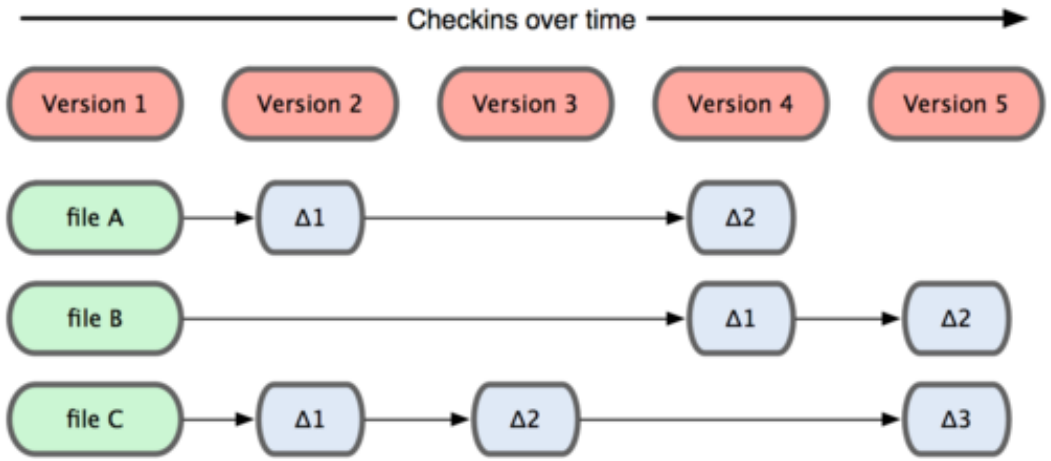


(Git, Mercurial)

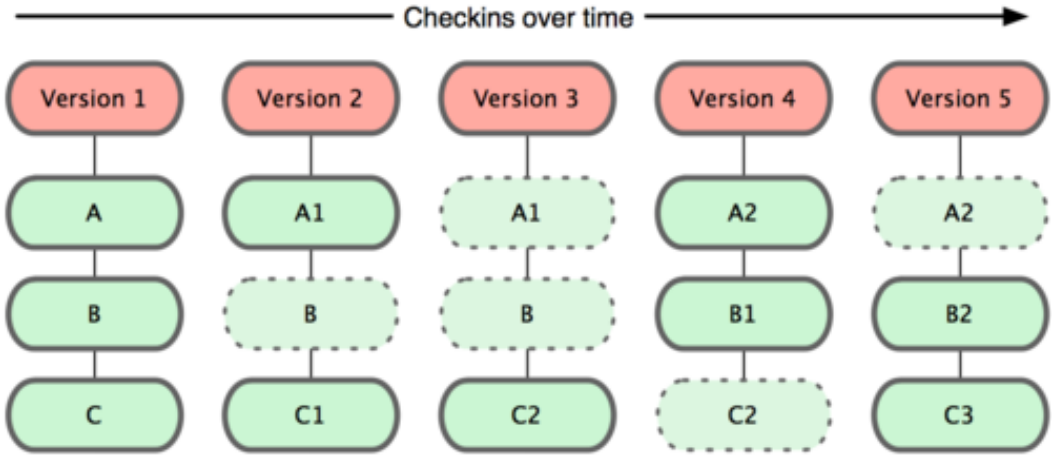
Result: Many operations are local

Git Takes Snapshots

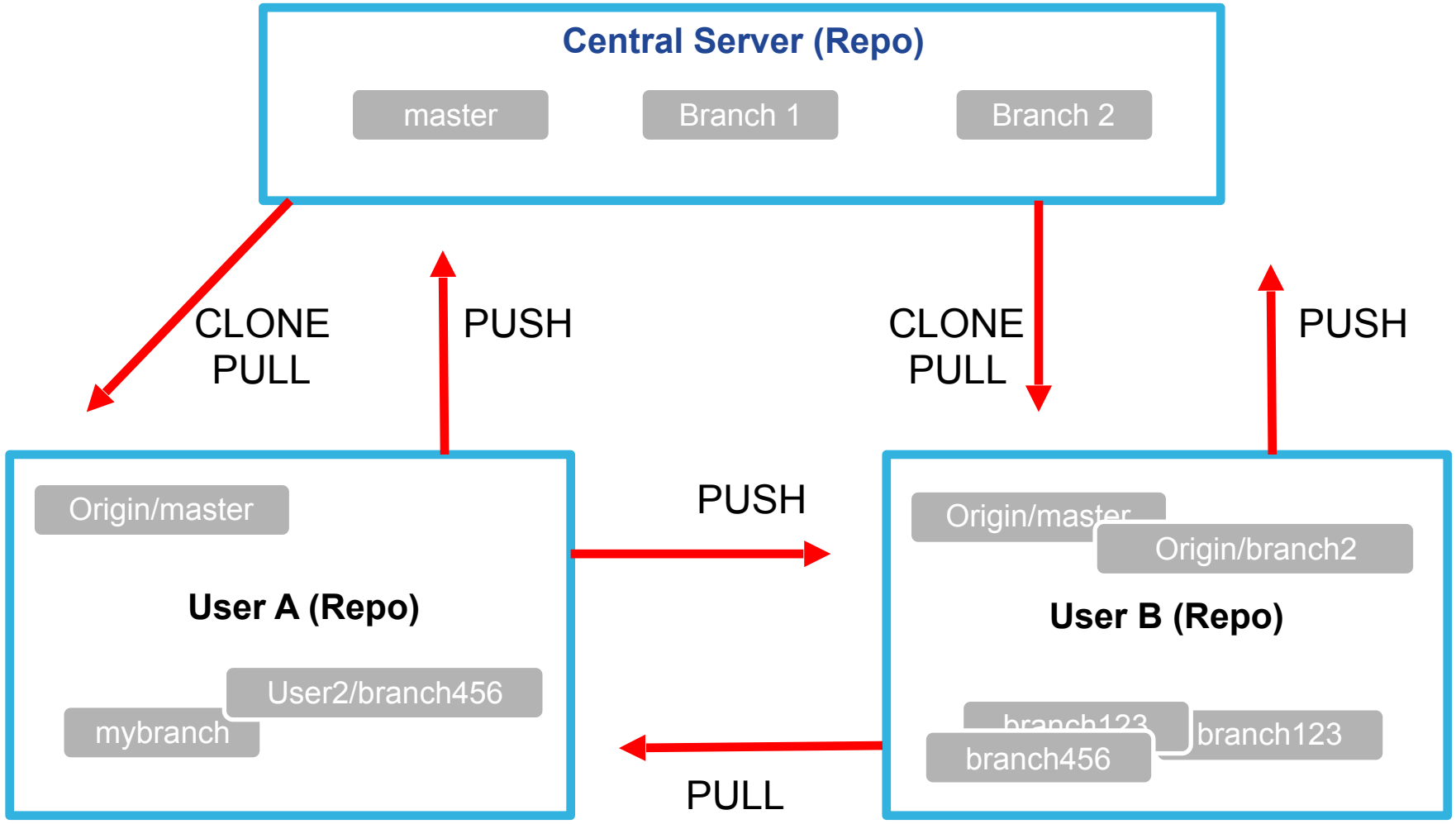
SUBVERSION



GIT



How Git Does It

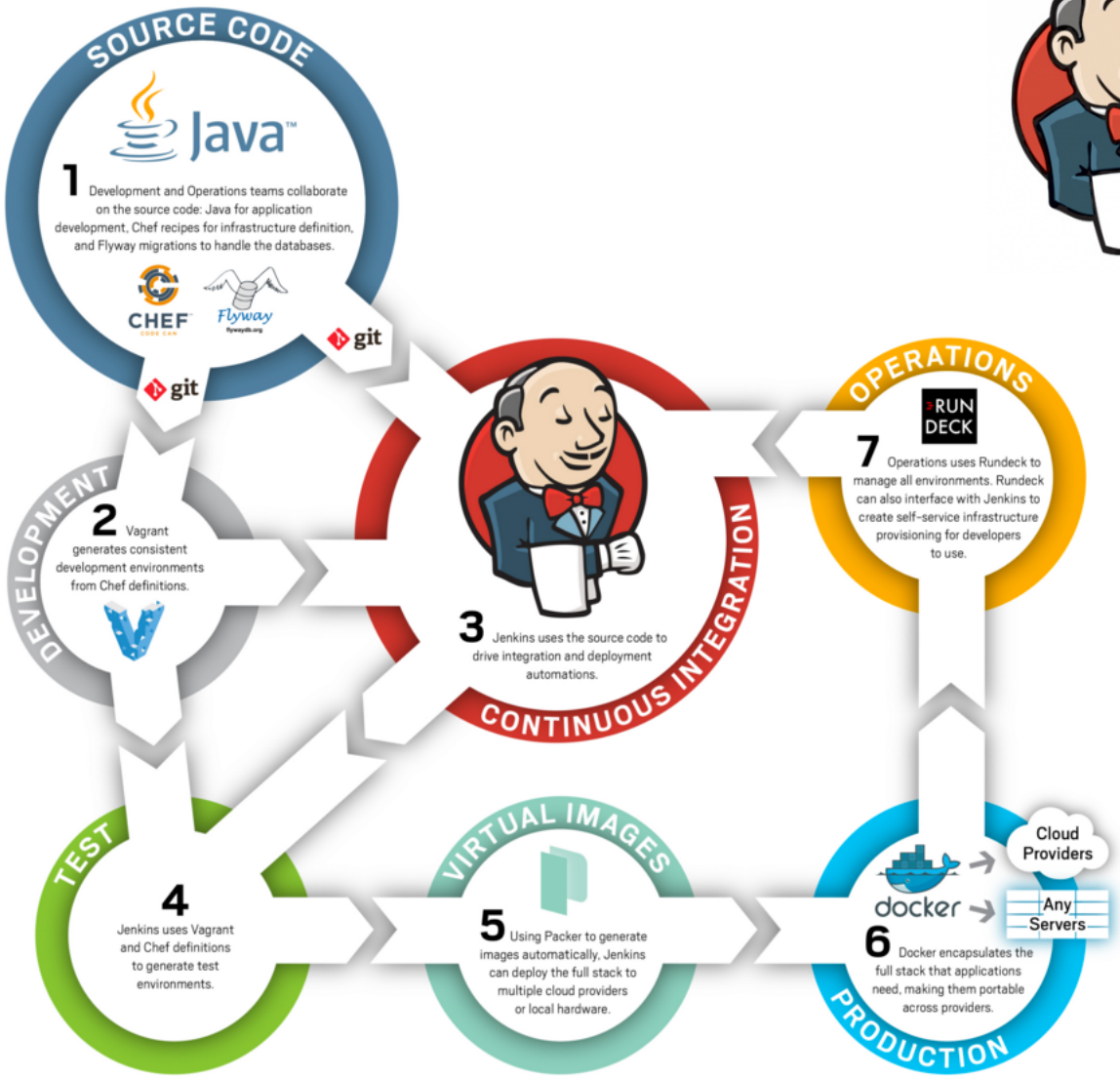


Jenkins

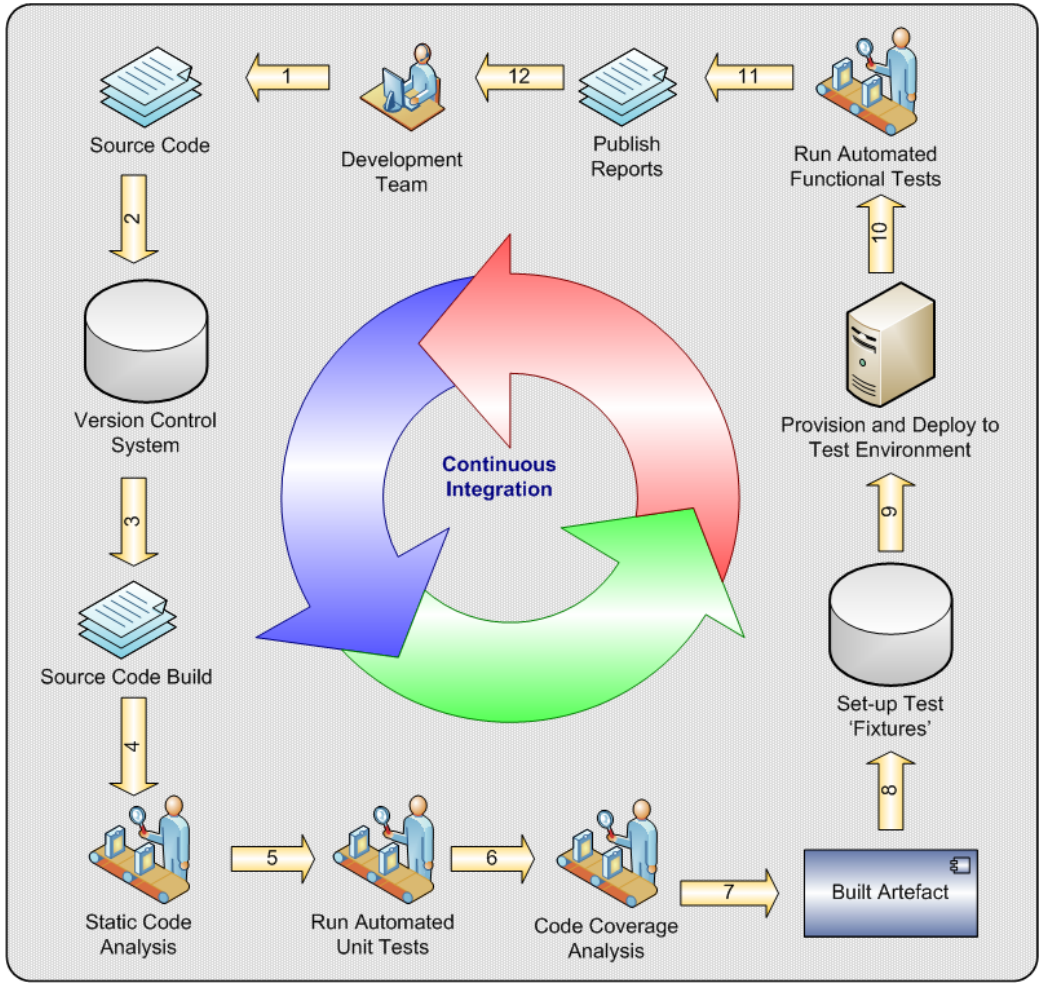
Continuous integration server:

- Coordinate “jobs” to automate the building of applications and environments
- Scales using master/slave architecture
- Integrates into all major DevOps and code build tools

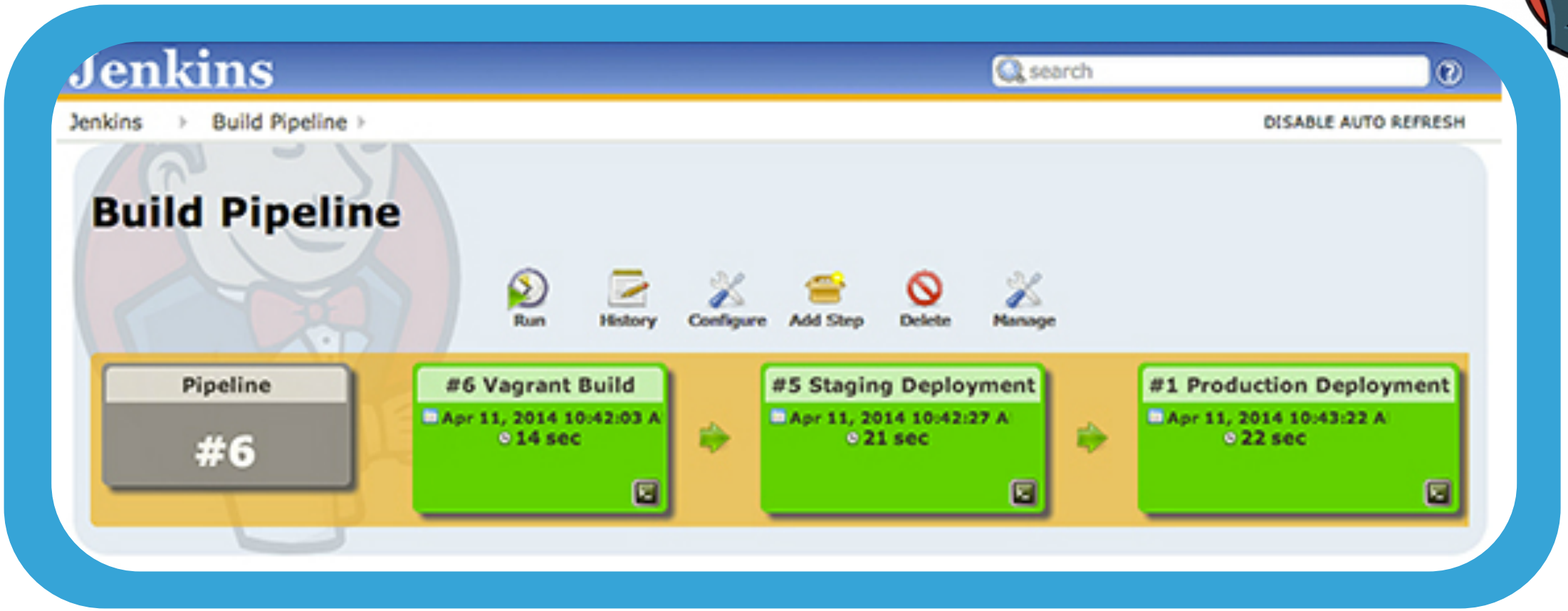
DevOps practices accelerate the rate of releases by increasing collaboration and reducing friction between developers and IT operations professionals. Open source tools help you to define good DevOps strategies and implement continuous deployment for projects of any size.



Typical CI Workflow using Jenkins



Jenkins Pipelines



Vagrant

- Provisioning tool for Dev and Test environments
- Why use Vagrant?
  - Quick
  - Easily replicate production on a Dev box
- Many use cases for developers, operations as well as Q/A

Vagrant: Basics

- Command line utility
- Uses a vagrant file to describe the virtual environment you want to set up
- Uses a simple abstraction from underlying complexities:
  - vagrant up
  - vagrant ssh
  - vagrant halt
  - Etc.

Vagrant: Basics (Cont.)

- The Box:
  - Basic building block
  - Uses shared “repo”
- Providers:
  - Supports AWS, VirtualBox, Vmware, etc.

Vagrant: When

- Need to easily provision development environments

Vagrant: Pros

- Initial set up to get running is quick and easy
- Wide availability of boxes to provision your environment

Vagrant: Cons

- Mysterious performance issues
- Some features on certain platforms (cough windows) can be a pain to implement

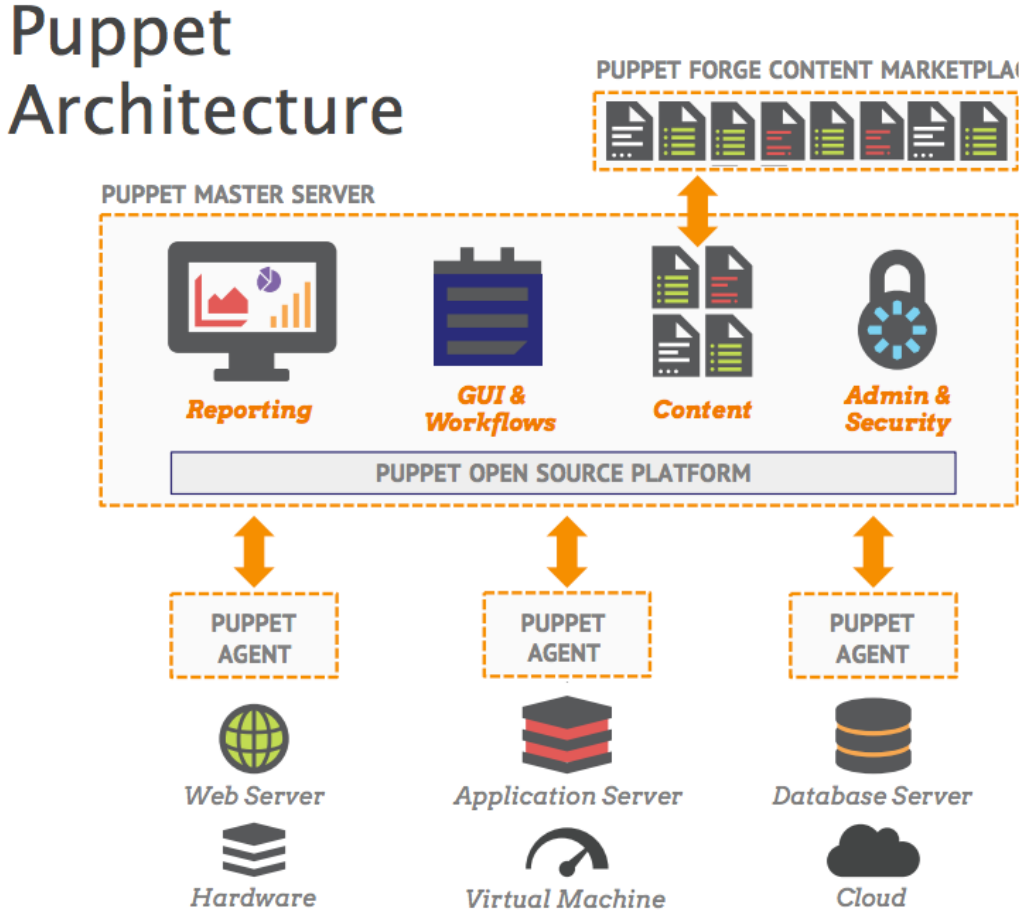
Puppet: What is Puppet?

- Think of it as a language
- Describe state, not steps
- Paint a picture of your ideal and most clean system

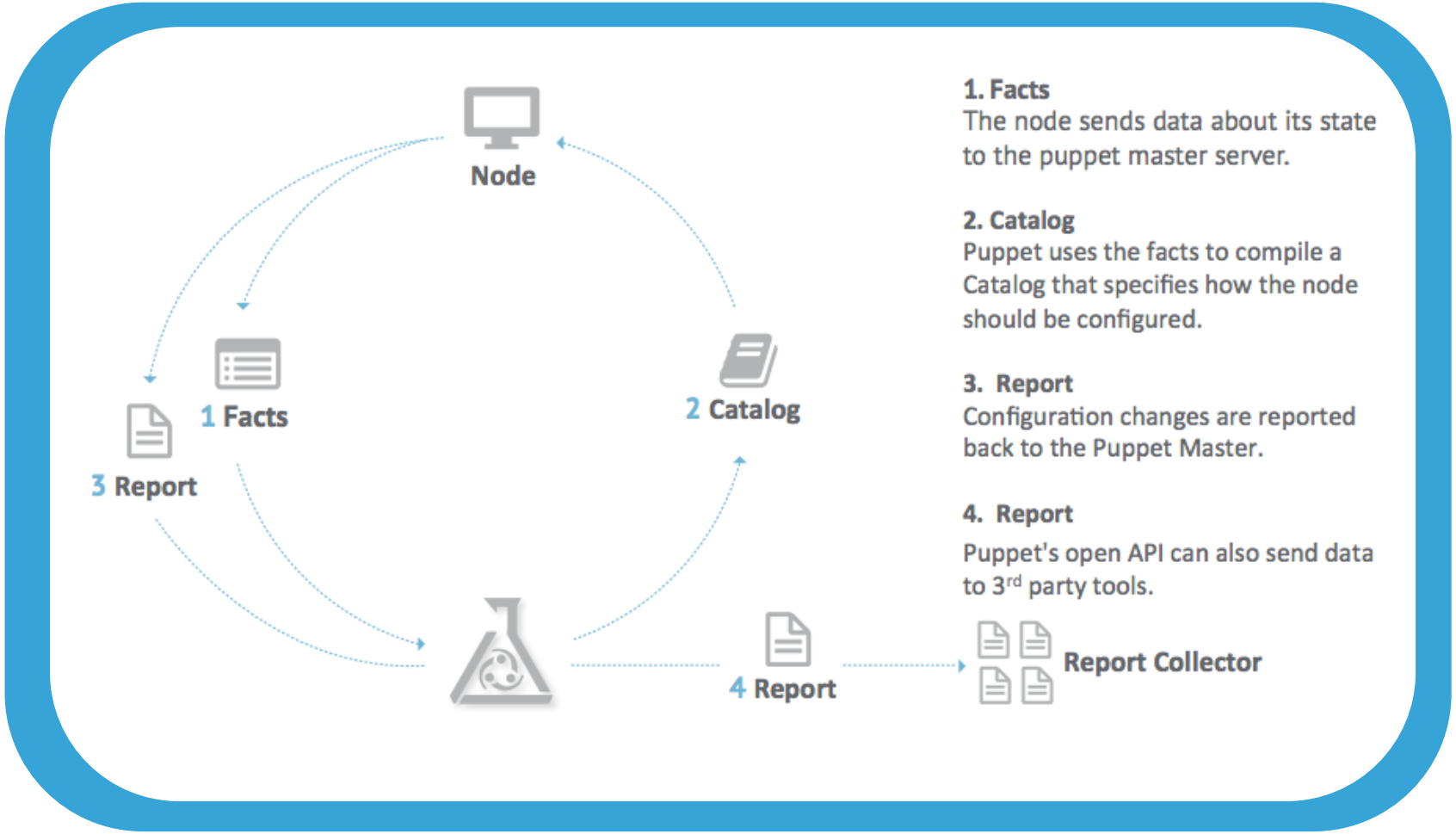
Puppet

- Automation of System Administration tasks
- Uses a declarative language to automate mundane admin tasks
- Seen as the tool of choice for pure configuration management
- Written in Ruby

Puppet: Architecture



Puppet: Puppet Run



Puppet: When

- Stability and maturity are key factors
- Good for large organizations with a heterogeneous environment and breadth of skills on the team

Puppet: Pros

- Mature interface and runs on nearly every OS
- Simple installation and setup
- Complete Web UI in this space
- Strong reporting capabilities
- Well-established support community

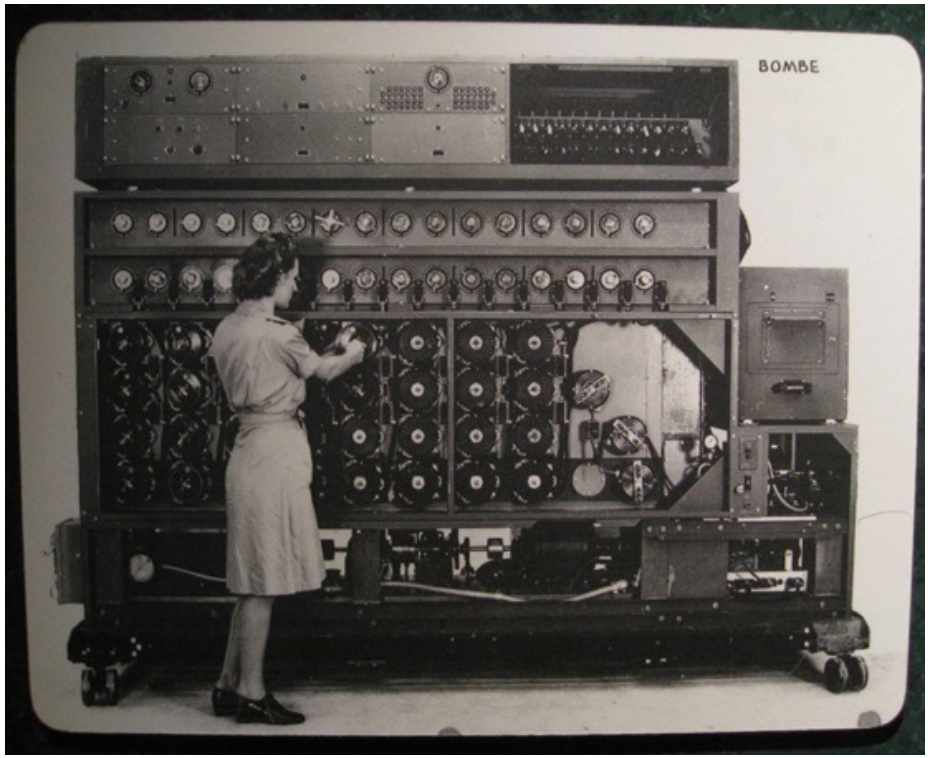
Puppet: Cons

- Advanced tasks will lead you to use the CLI, which is Ruby-based
- Due to the DSL and a non-simplistic design, a Puppet code base can grow large, complex, and will be hard for new team members to become productive on quickly
- Model-driven Pure-Ruby version support is being scaled back
- Versus code-driven approach

Chef

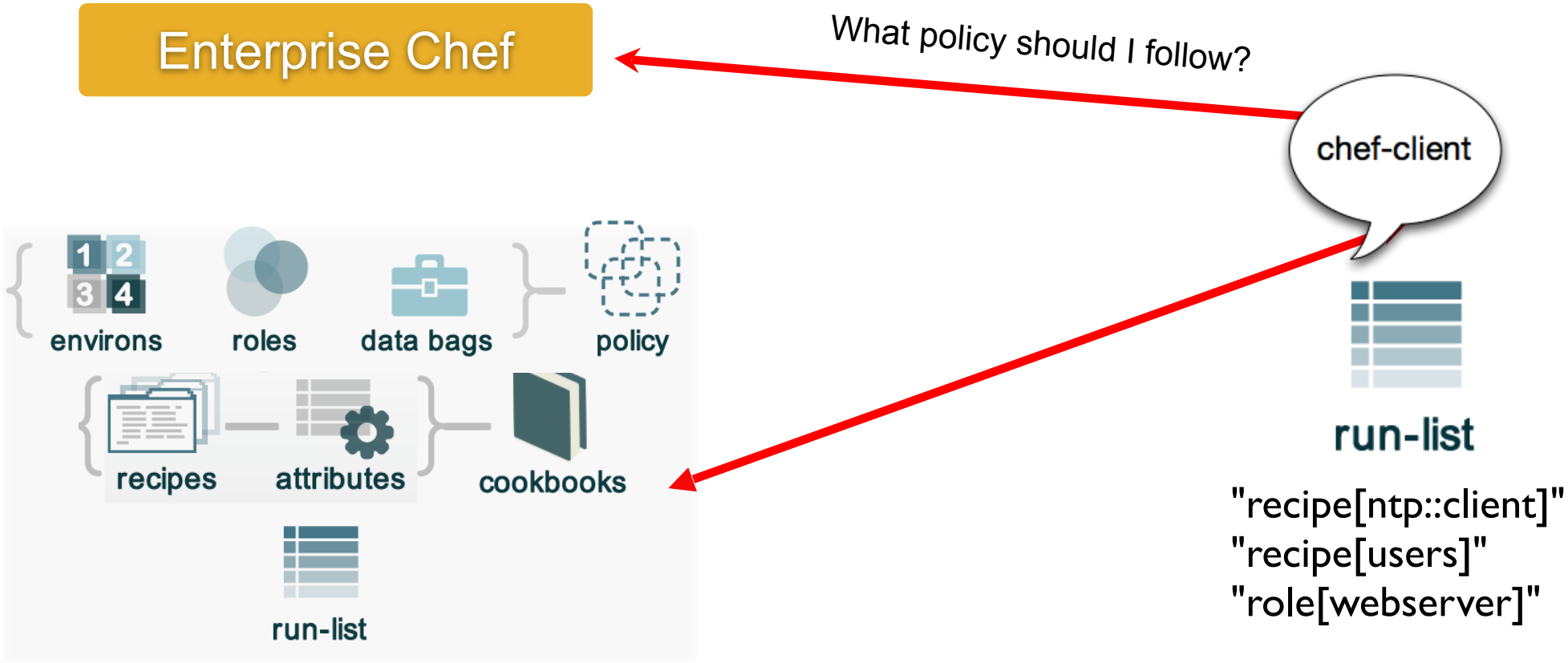
- Chef is the ninja of DevOps tools
- Comprehensive utility for:
  - provisioning
  - managing
  - automating entire infrastructures
- Uses a client/server architecture to facilitate application of “recipes” to infrastructure components
- Written in Ruby

Chef: Concepts



- Node
- Role
- Resource
- Recipe
- Cookbook
- Runlist

Chef: Run List



Chef: When/Why

- Organizations looking for a more mature solution for a heterogeneous environment
- Good for development-focused teams

Chef: Cons

- Not simple—can lead to large code bases and complex environments
- Doesn’t support push functionality
- Steep learning curve

Chef: Cons

- “Knife” tool eases installation burdens
- Large collection of modules and configuration recipes
- Heavy focus on Git gives it strong version control capabilities
- Code-driven approach gives you more control configurations

Saltstack

- CLI-based tool that can be set up as a master-slave model or a non-centralized model.
- Python
- Push method of communication with clients
- Provides for grouping of clients and configuration templates

Saltstack: When

- Scalability and resiliency are a big concern.
- Oriented to system administrators due to its usability

Saltstack: Pros

- After setup, fairly straightforward
- DSL is robust with features
- I/O and configs are consistent-> YAML.
- Strong community
- High scalability and resiliency in the master model
- Transparency is excellent. Easy to see what's happening internally

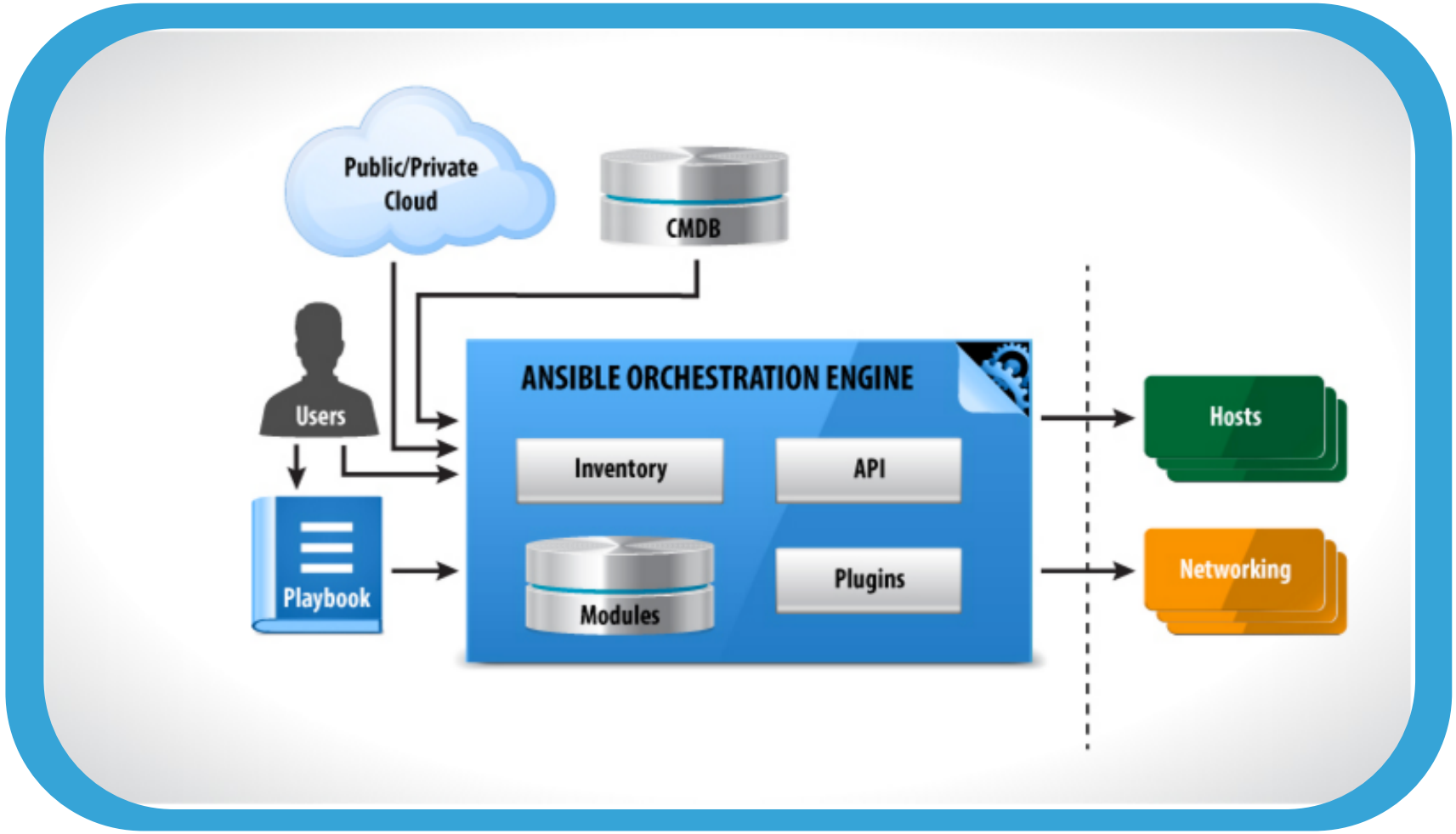
Saltstack: Cons

- Documentation is weak
- Incomplete web UI
- Non Linux OS support weak
- Difficult to set up and to pick up for new users.

Ansible

- Agentless configuration management tool
- Written in Python
- Multi-node (like chef)
- Seeks to be simple
- Similar to Vagrant

Ansible: Architecture



Ansible: Key Concepts

- Module
- Playbook

Ansible: When/Why

- Getting up and running quickly
- Ease is important to you
- You don’t want to install agents on remote nodes or managed servers
- It’s good if your focus is more on the system admin side.

Ansible: Pros

- SSH-based, so it doesn’t require installing any agents on remote nodes.
- Easy learning curve thanks to the use of YAML.
- Playbook structure is simple and clearly structured.
- Has a variable registration feature that enables tasks to register variables for later tasks
- Much more streamlined code base than some other tools

Ansible: Cons

- Less powerful than tools based in other programming languages.
- Does its logic through its DSL, which means checking in on the documentation frequently until you learn it
- Variable registration is required for even basic functionality, which can make easier tasks more complicated
- Introspection is poor. Difficult to see the values of variables within the playbooks
- No consistency between formats of input, output, and config files
- Struggles with performance speed at times.

Teraform

- Higher level abstraction of CM
- Written in G (golang)
- Focuses on data center wide operations
- Integrates with other CM tools like Chef and Puppet
- From the makers of Vagrant

Teraform: Configuration

Like Vagrant:

- Command line tool
- Uses text files to describe the infrastructure

Teraform: When

- The planning phase plain and simple:
  - Killer feature
- When the holes in a young product are not in areas you need support. (Missing resources)

Teraform: Pros

- Supports many cloud providers and can run on your local machine
- Vendor neutral
- Separate planning and execution phase—killer feature
- Young—stability can be an issue

Teraform: Cons

- Limited support for many AWS components
- Issues with state management—state file sharing between developers is chaos
- DSL is young and missing some functionality

Cloud Foundry

- Provision operating systems and middleware.
- Manage network security safe guards to prevent security threats.
- Manage application connections to external sources including databases and legacy middleware.
- Provides 4 levels of HA, as well as built in load balancing for scale.

Cloud Foundry (Cont.)

- Supports multi-tenant environments—each line of business can operate with a discrete quota and isolated system access.
- Provision next generation data services including NOSQL databases, traditional databases and Hadoop clusters.
- We provide horizontal and vertical scaling for the underlying IaaS so that you can scale your infrastructure in lock step with your Business.
- Provides a built-in log aggregation service.

Cloud Foundry: Open Source Capabilities

CLOUD FOUNDRY™

Automatic AppServer & OS Configuration with Buildpacks ("just push your app")

Application Containerization & Cluster Scheduling

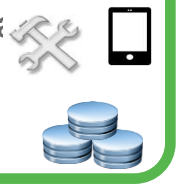
Application Network Security Groups

Application to Services Binding and Access

App Health Mng, Load Balancing, Rapid Scaling, Availability Zones

Policy, Identity and Roles Management

Native & Extended Data Mobile and Platform Services



IaaS Provisioning, Scaling & Configuration

Basic logging as a service

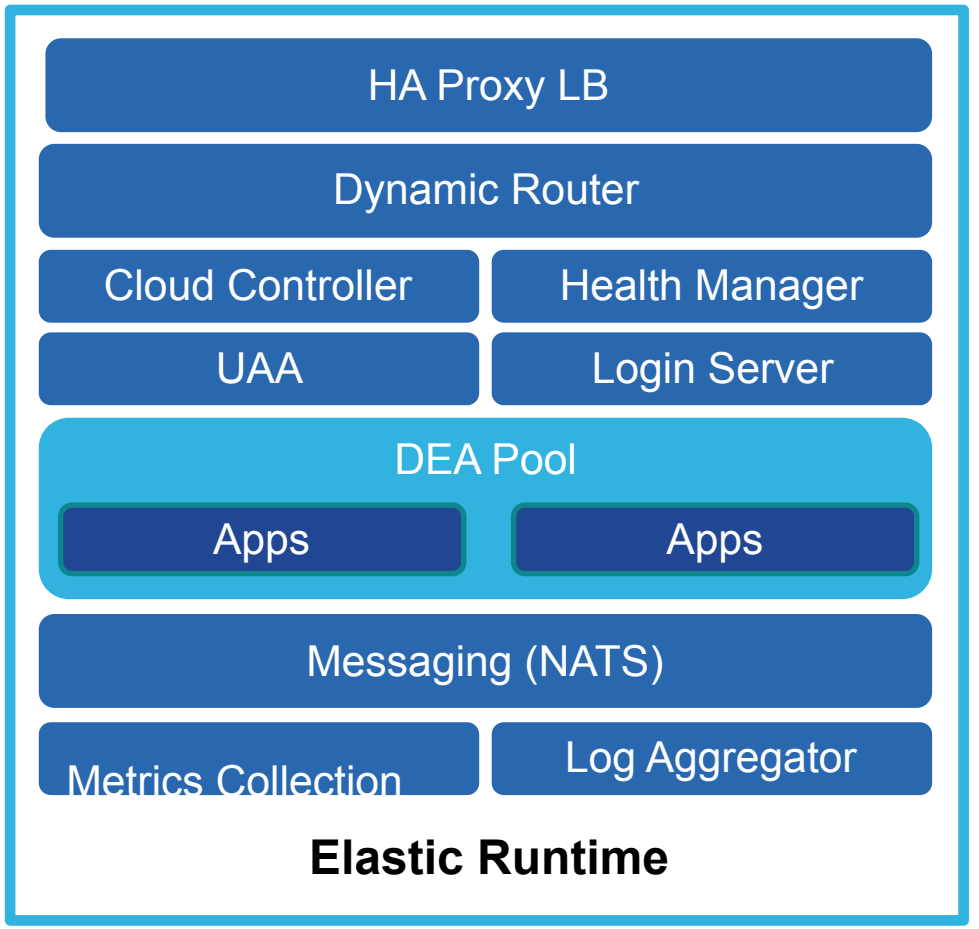
Deploy, Operate, Update & Scale with minimal downtime

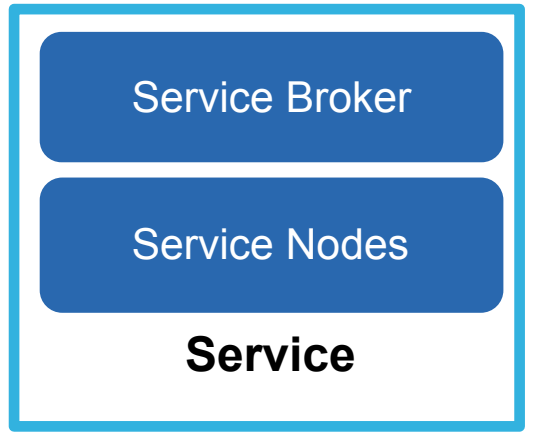
vmware

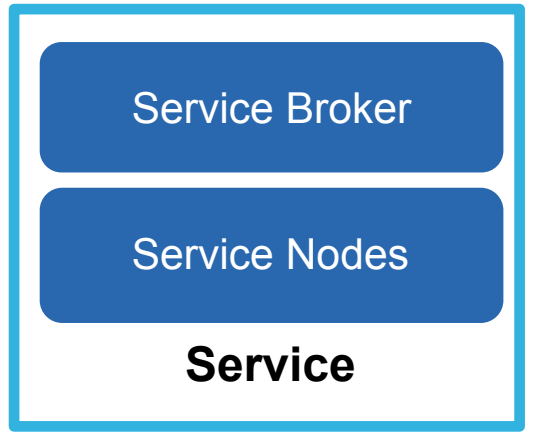
amazon web services™

openstack™

Cloud Foundry: Components







Cloud Foundry: When

- You are running on AWS
- Significant number of ready to go resource definitions

Cloud Foundry: Pros

- Support for all AWS components
- No state management issues
- Templates written in son and very easy to understand
- Very stable

Cloud Foundry: Cons

- AWS only and is a hosted service
- Can makes changes without prior notice - lack of transparency
- Issues with rollback mechanism

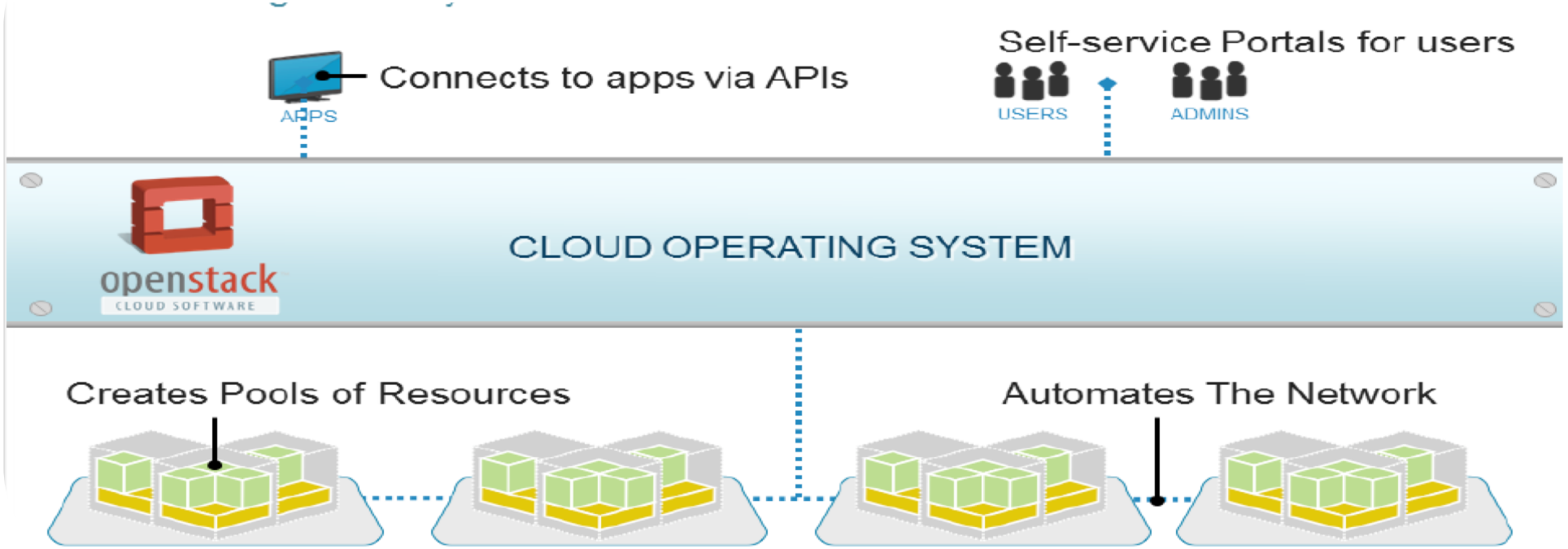
OpenStack

- Software platform for cloud computing
- Infrastructure-as-a-service (IaaS)
- The software platform consists of interrelated components that control hardware pools of processing, storage, and networking resources throughout a data center.

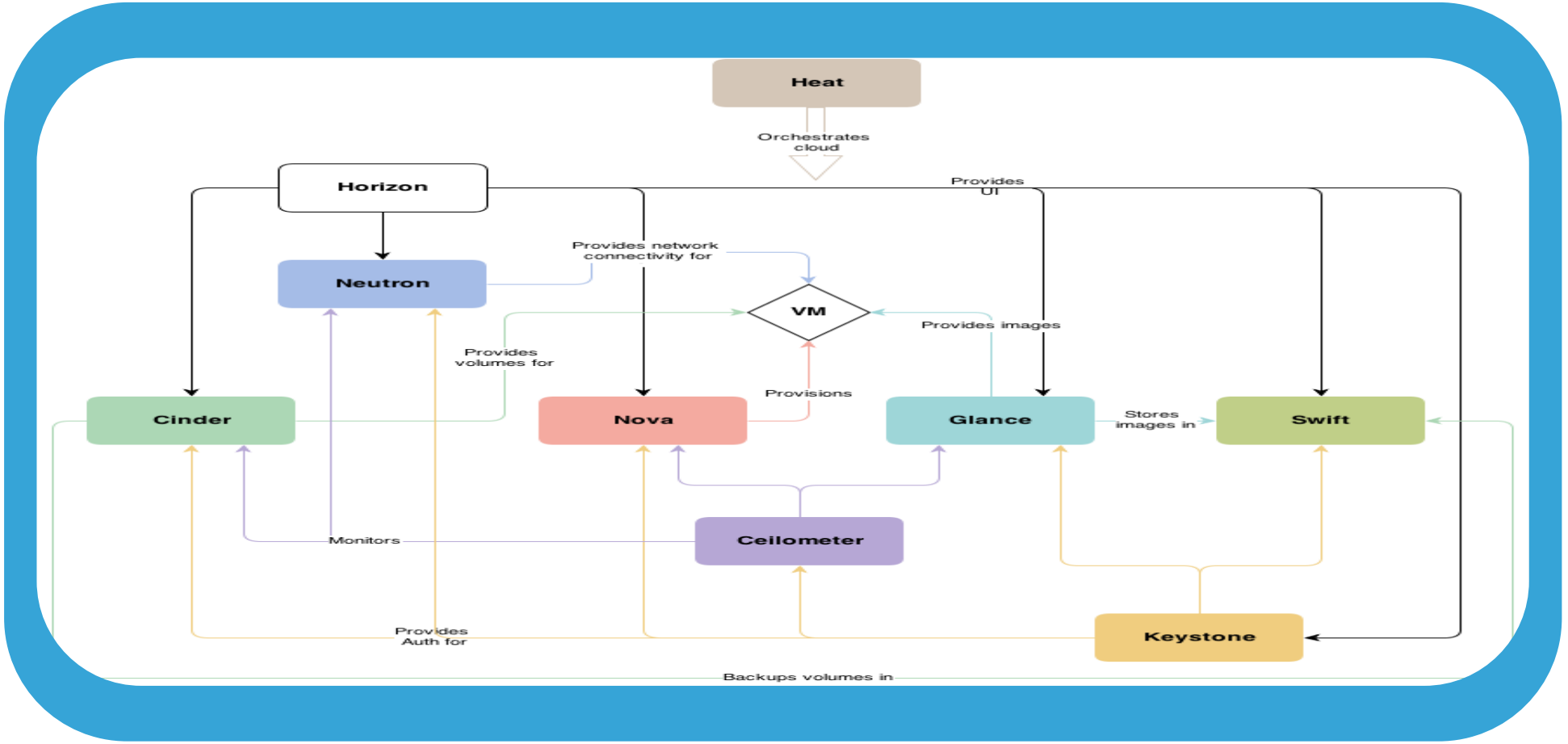
Automation and Orchestration of IT Resources

Solution: OpenStack, the Cloud Operating System

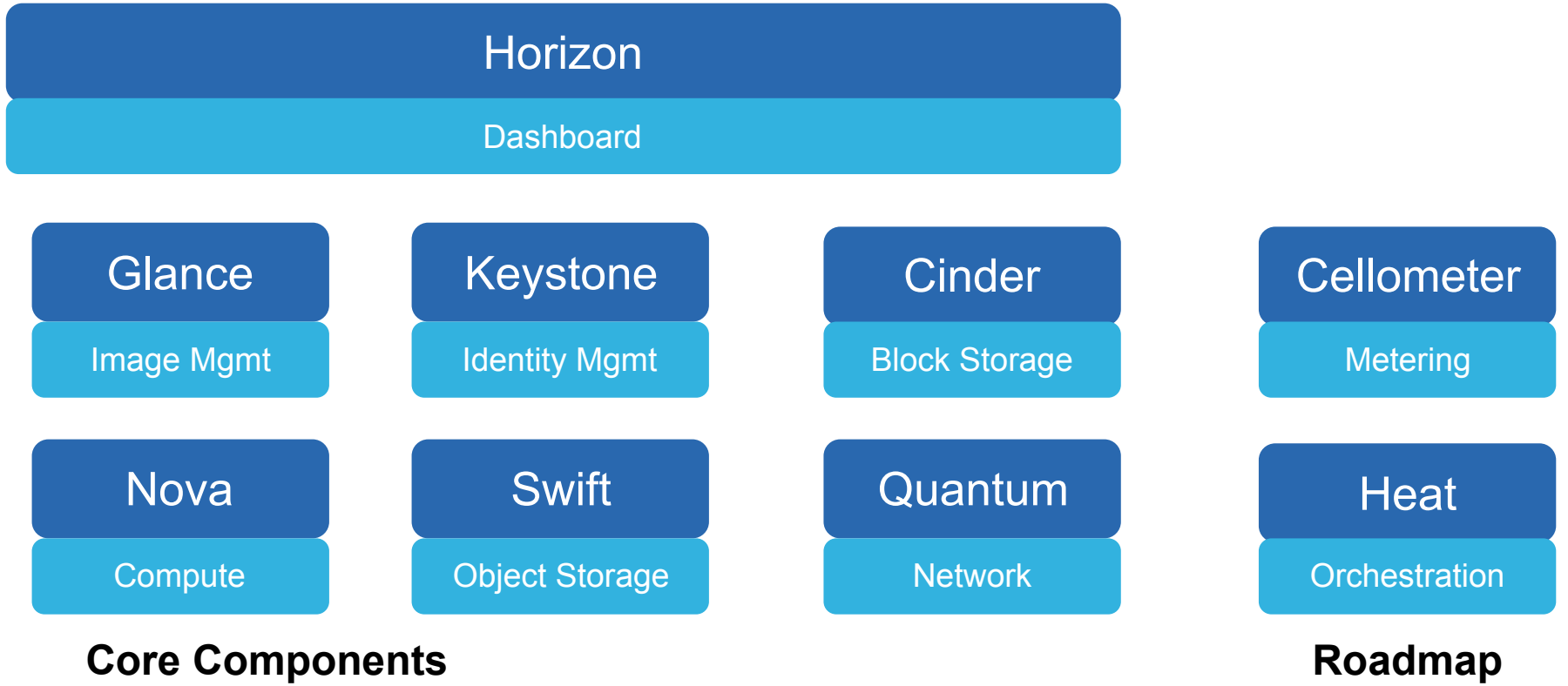
A new management layer that adds automation and control



In a Loosely Coupled Architecture



By Leveraging Various Open Source Projects



Rackspace Private Cloud Reference Architecture