

# What's New with the Cloud?

A quick look at the evolution  
and possible future of cloud computing

*A view from the past 10 years of working on and using cloud technology*

Dennis Gannon, Professor Emeritus, SOICE

# Outline


- Defining a cloud
  - Public, Private, Hybrid, Research, Academic, ...
- The software
  - The evolution of services and cloud-native programming models
- The data centers
  - From racks of PCs to planetary-scale special supercomputers
- The future
  - AI and the Edge



# Defining the cloud

- Let's ask Google, Bing and Alexa: *"What is the cloud?"*

cloud

[klaʊd] 

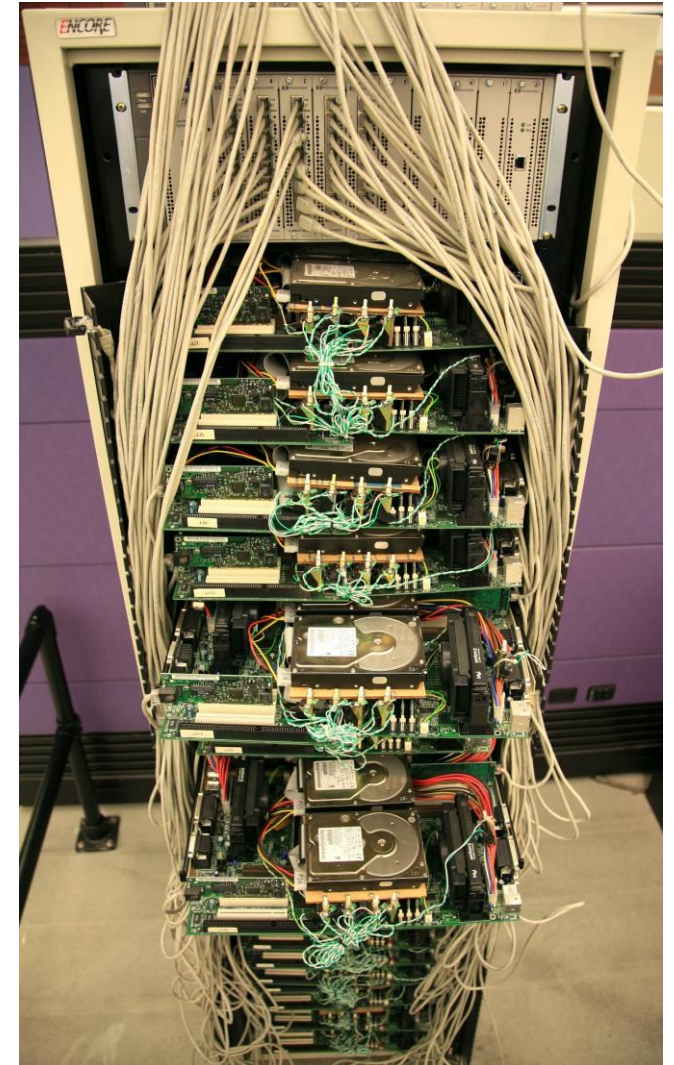
NOUN

1. a visible mass of condensed water vapor floating in the atmosphere, typically high above the ground:  
"the sun had disappeared behind a cloud" · [\[more\]](#)
2. a state or cause of gloom, suspicion, trouble, or worry:  
"the only cloud to appear on the horizon was Leopold's unexpected illness" · [\[more\]](#)
3. a network of remote servers hosted on the Internet and used to store, manage, and process data in place of local servers or personal computers:



# Better answers?

- The “Cloud” began life as the place Google did the analysis of all of its data and where they stored their index (1998)
- The gold rush for “on-line” drove data centers.
  - Google, Microsoft, Amazon, Yahoo! ...
  - Services: web search, games and email
- The breakthrough in 2006:
  - Amazon AWS S3 blob storage as a service
  - AWS EC2 virtual machines as a service.
  - “the combined technologies of S3, EC2 and Mechanical Turk represented the culmination of 11 years of web-scale computing” – Jeff Bezos 2006
  - Microsoft Azure 2008 ... released 2010.



first iteration of Google production servers

# Types of Clouds

- “public clouds” vs “private” vs “science private clouds”
  - Public = anybody with a credit card has access. (aka commercial cloud)
  - Private = restricted to a special group of users. (aka Community Cloud or Academic Cloud)
  - (In Europe these terms are often reversed based on ownership.)
- Examples:
  - Amazon Web Services (AWS) - 40% of all cloud resources on the planet.
  - Microsoft Azure – about 1/3 of AWS but growing
  - Google Cloud – third place
  - IBM Bluemx - growing
  - NSF JetStream – an OpenStack private cloud for US science researchers.
- There are *many* more clouds.
  - Public: Salesforce, DigitalOcean, Rackspace, 1&1, UpCloud, CityCloud, CloudSigma, CloudWatt, Aruba
  - Private Research Clouds: Aristotle, Bionimbus, Chameleon, RedCloud, indigo-datacloud, EU-Brazil Cloud, European Open Science Cloud
- What are the pros and cons of public vs private

# Pros & Cons of Public vs Private Cloud

- Public cloud pros

- Massive scale
- Huge and growing list of services
- Highly competitive on pricing due to economies of scale
- Physical Security is strong
- Freedom from managing hardware
- Hardware constantly upgraded

- Cons

- Rules prohibit moving data to cloud
- Funding models may make it hard to use
- Fear of “vendor Lock-In”
- Securing your data is still up to you

- Private cloud pros

- May be cheaper
- You can keep it off the Internet so data can be very safe.
- You can optimize your own hardware
- You control everything
- Fun for systems research

- Cons

- You are responsible for everything
- Not as many high level services
- May not really be cheaper
- You manage physical and system security

# Azure and AWS Now Global Scale





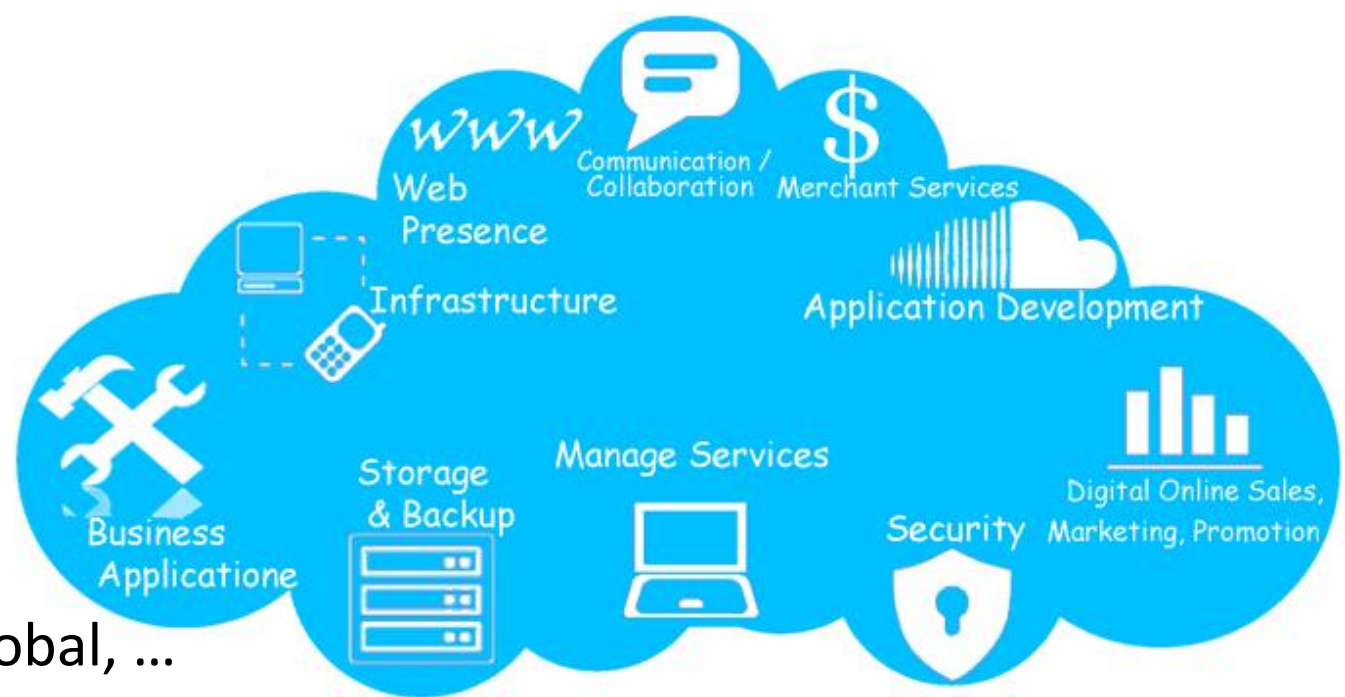
# Software

The true essence of “cloudiness”

# Cloud Services

- ***on-demand access to***

- Data storage: blob, file, unstructured, SQL, global, ...
- Raw computer: VM, cluster, GPU
- App services: Basic web hosting, mobile app backend
- Streaming data: IoT data streams, web log streams, instruments
- Security services: user authentication, delegation of authorization, privacy, etc.
- Analytics: database, BI, app optimization, stream analytics
- Integrative: networking, management services, automation



# Clouds are all about Services - Here is Azure

## Platform Services

### Security and Management

- Portal
- Active Directory
- Multi-factor Authentication
- Automation
- Key Vault
- Store/Marketplace
- VM Image Gallery and VM Depot

### Compute

- Cloud Services
- Service Fabric
- Batch
- Remote App

### Web and mobile

- Web Apps
- API Apps
- API Management
- Mobile Apps
- Logic Apps
- Notification Hubs

### Developer services

- Visual Studio
- Azure SDK
- Team Project
- Application Insights

### Hybrid Operations

- Azure AD Connect Health
- AD Privileged Identity Management
- Backup
- Operational Insights
- Import/Export
- Site Recovery
- StorSimple

### Integration

- Storage Queues
- Biztalk Services
- Hybrid Connections
- Service Bus

### Analytics and IoT

- HDInsight
- Machine Learning
- Data Factory
- Event Hubs
- Stream Analytics
- Mobile Engagement

### Data

- SQL Database
- SQL Data Warehouse
- Redis Cache
- Search
- DocumentDB
- Tables

### Media and CDN

- Media Services
- Content Delivery Network (CDN)

## Infrastructure Services

### Compute

- Virtual Machine
- Containers

### Storage





- BLOB Storage
- Azure Files
- Premium Storage

### Networking







- Virtual Network
- Load Balancer
- DNS
- Express Route
- Traffic Manager
- VPN Gateway
- Application Gateway

# Amazon Web Services





## Compute

-  **EC2**  
Virtual Servers in the Cloud
-  **EC2 Container Service**  
Run and Manage Docker Containers
-  **Elastic Beanstalk**  
Run and Manage Web Apps
-  **Lambda**  
Run Code in Response to Events




## Storage & Content Delivery

-  **S3**  
Scalable Storage in the Cloud
-  **CloudFront**  
Global Content Delivery Network
-  **Elastic File System** **PREVIEW**  
Fully Managed File System for EC2
-  **Glacier**  
Archive Storage in the Cloud
-  **Import/Export Snowball**  
Large Scale Data Transport
-  **Storage Gateway**  
Integrates On-Premises IT Environments with Cloud Storage

## Database

-  **RDS**  
Managed Relational Database Service
-  **DynamoDB**  
Predictable and Scalable NoSQL Data Store
-  **ElastiCache**  
In-Memory Cache
-  **Redshift**  
Managed Petabyte-Scale Data Warehouse Service








## Networking

-  **VPC**  
Isolated Cloud Resources
-  **Direct Connect**  
Dedicated Network Connection to AWS
-  **Route 53**  
Scalable DNS and Domain Name Registration





## Developer Tools

-  **CodeCommit**  
Store Code in Private Git Repositories
-  **CodeDeploy**  
Automate Code Deployments
-  **CodePipeline**  
Release Software using Continuous Delivery





## Management Tools

-  **CloudWatch**  
Monitor Resources and Applications
-  **CloudFormation**  
Create and Manage Resources with Templates
-  **CloudTrail**  
Track User Activity and API Usage
-  **Config**  
Track Resource Inventory and Changes
-  **OpsWorks**  
Automate Operations with Chef
-  **Service Catalog**  
Create and Use Standardized Products
-  **Trusted Advisor**  
Optimize Performance and Security

## Security & Identity

-  **Identity & Access Management**  
Manage User Access and Encryption Keys
-  **Directory Service**  
Host and Manage Active Directory
-  **Inspector** **PREVIEW**  
Analyze Application Security
-  **WAF**  
Filter Malicious Web Traffic






## Analytics

-  **EMR**  
Managed Hadoop Framework
-  **Data Pipeline**  
Orchestration for Data-Driven Workflows
-  **Elasticsearch Service**  
Run and Scale Elasticsearch Clusters
-  **Kinesis**  
Work with Real-time Streaming data








## Internet of Things

-  **AWS IoT** **BETA**  
Connect Devices to the cloud




## Mobile Services

-  **Mobile Hub** **BETA**  
Build, Test, and Monitor Mobile apps
-  **Cognito**  
User Identity and App Data Synchronization
-  **Device Farm**  
Test Android, Fire OS, and iOS apps on real devices in the Cloud
-  **Mobile Analytics**  
Collect, View and Export App Analytics
-  **SNS**  
Push Notification Service

## Application Services

-  **API Gateway**  
Build, Deploy and Manage APIs
-  **AppStream**  
Low Latency Application Streaming
-  **CloudSearch**  
Managed Search Service
-  **Elastic Transcoder**  
Easy-to-use Scalable Media Transcoding
-  **SES**  
Email Sending Service
-  **SQS**  
Message Queue Service
-  **SWF**  
Workflow Service for Coordinating Application Components

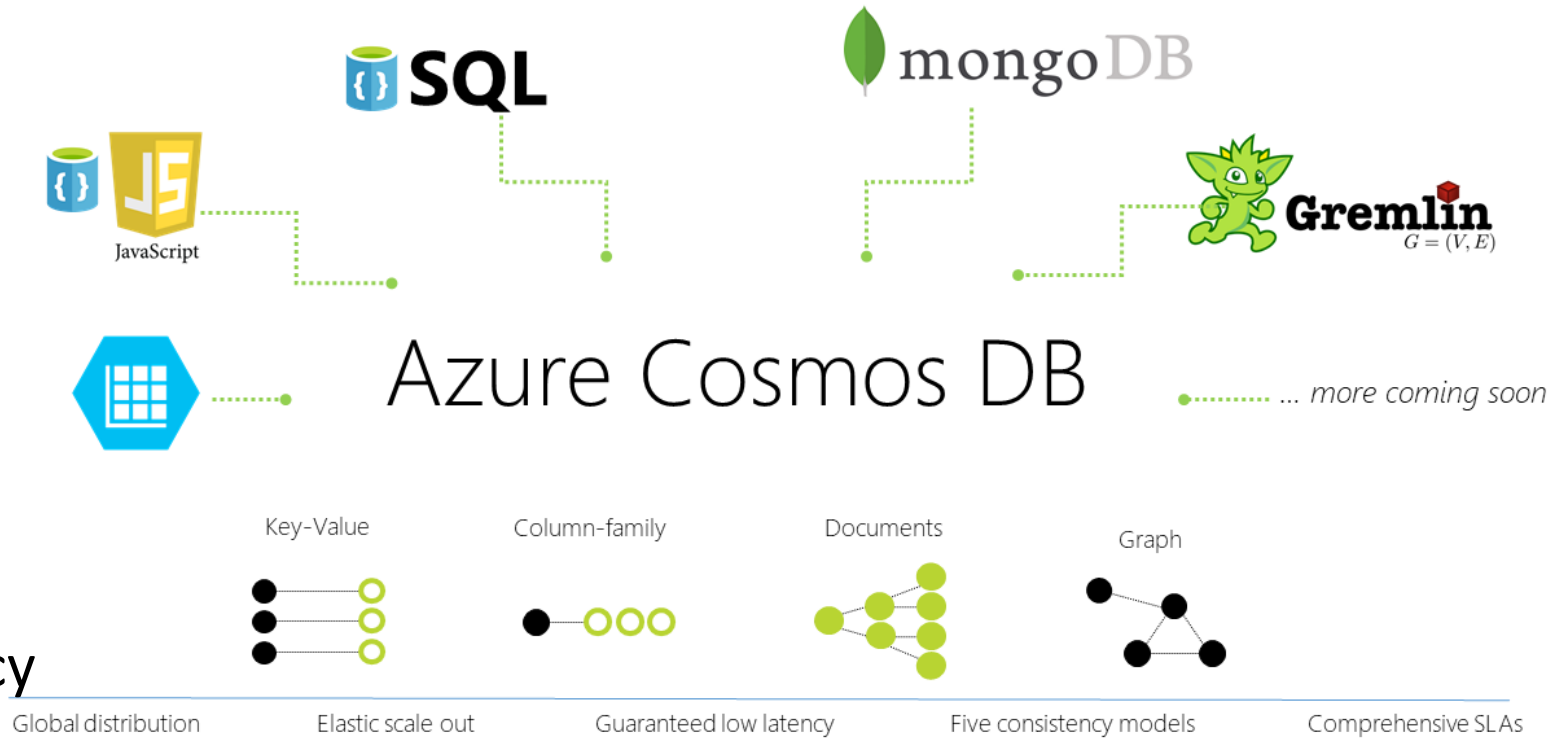
## Enterprise Applications

-  **WorkSpaces**  
Desktops in the Cloud
-  **WorkDocs**  
Secure Enterprise Storage and Sharing Service
-  **WorkMail** **PREVIEW**  
Secure Email and Calendaring Service

# A brief look at three big services

- Azure Cosmos Database

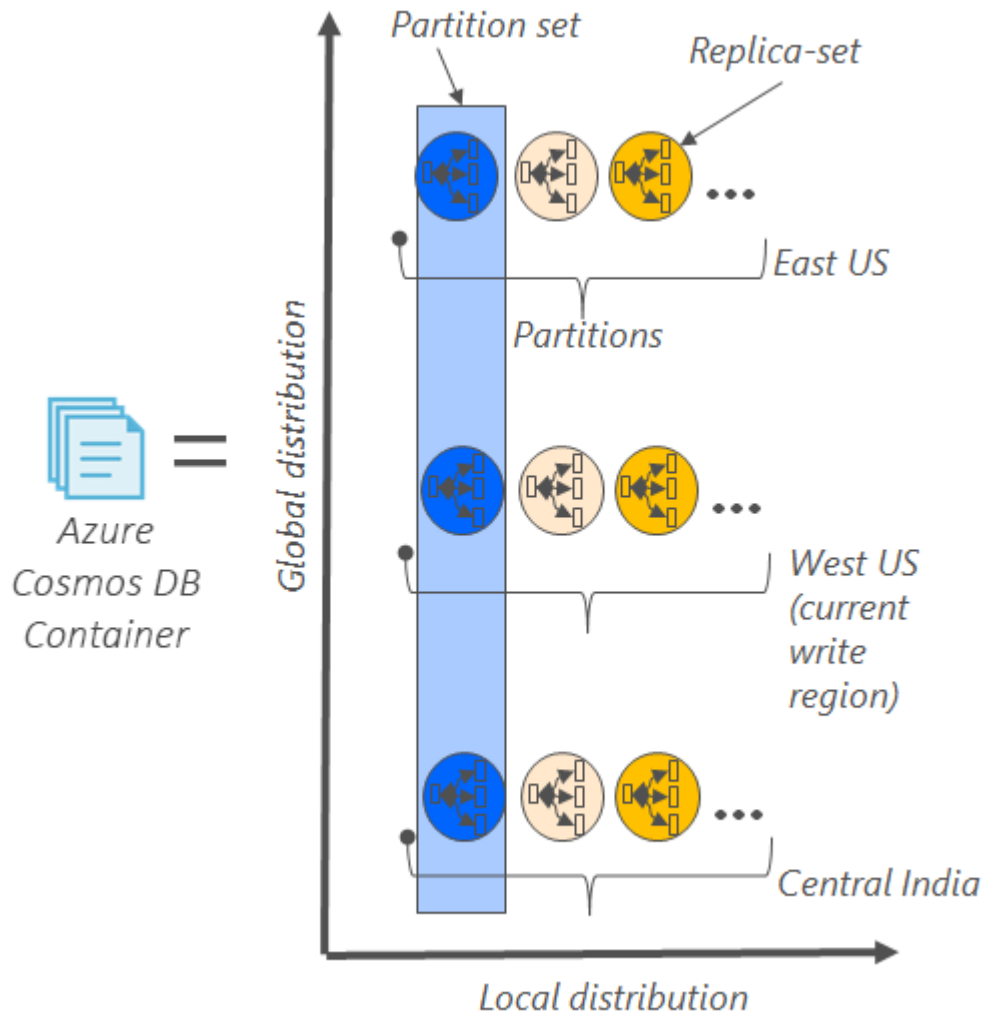
- 4 modes
  - Documents, key-value
  - Graph, NoSQL
- 5 consistency models
  - Eventual, consistent prefix
  - Session, bounded stateless
  - Strong consistency
- 99.9% less than 15ms latency
- Strong SLA
- Pay only for what you use
- Planet scale ....



A globally-distributed, multi-model database service



# Cosmos DB global and local distribution



Replicate data globally  
bookdocdb

Save Discard Manual Failover Automatic Failover

Click on a location to add or remove regions from your Azure Cosmos DB account.  
\* Each region is billable based on the throughput and storage for the account. [Learn more](#)

World map showing active regions (indicated by blue checkmarks).

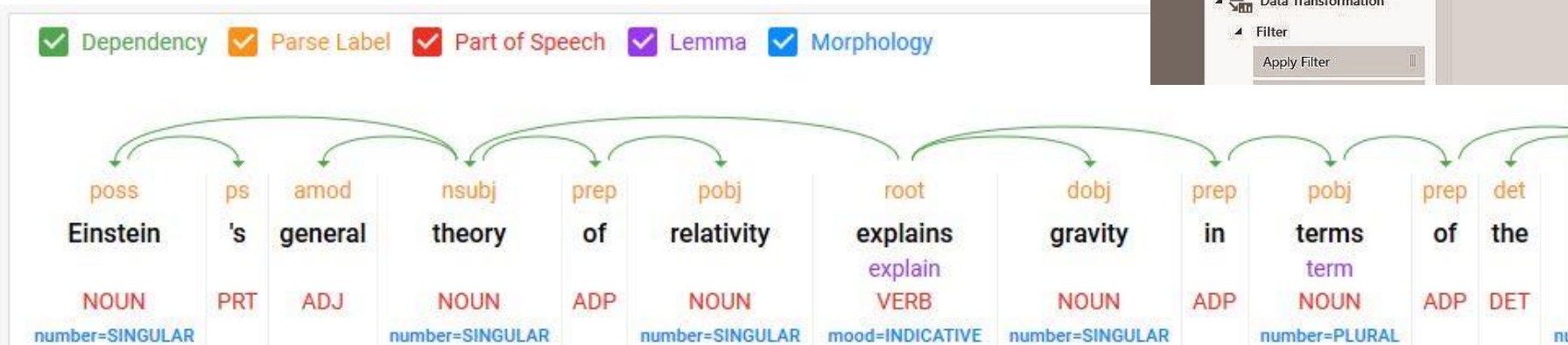
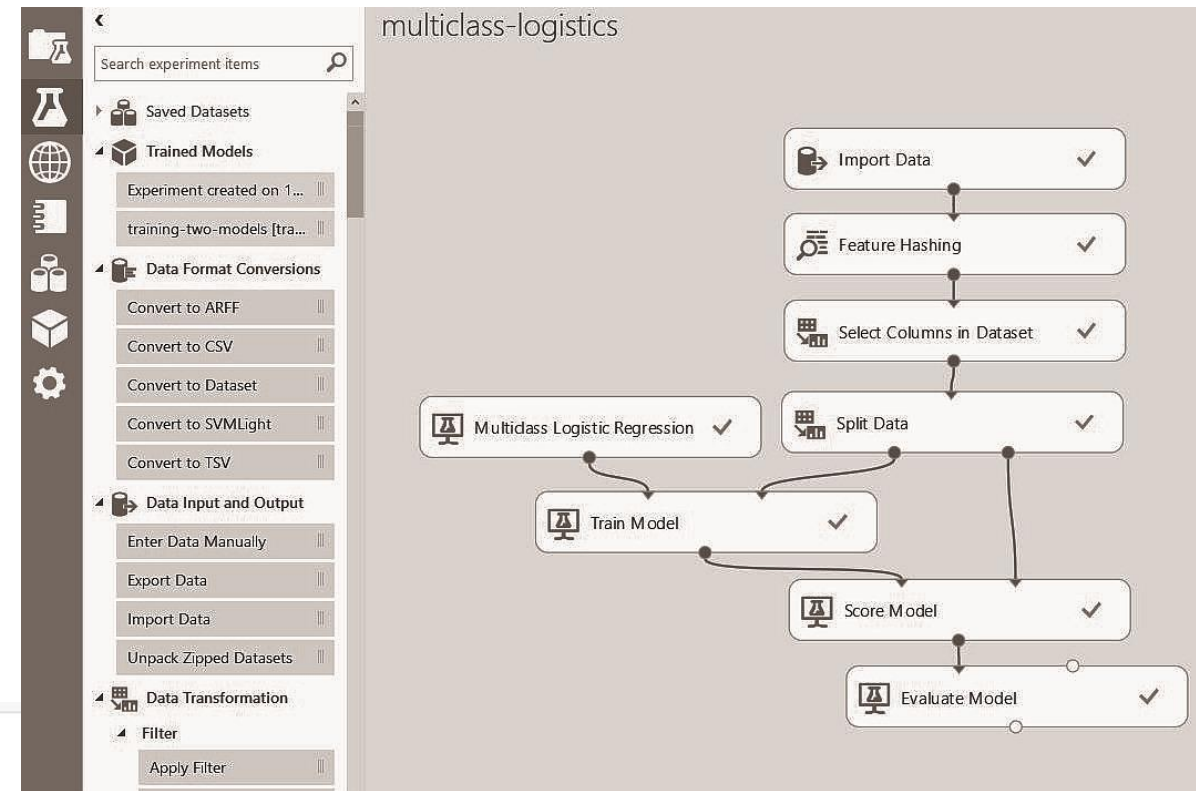
# Another Example: Amazon Kinesis

- A service to build data stream analytics
- Three components
  - Data ingest from external devices: Kinesis Streams, or Firehose
  - Kinesis Analytics
- Trivial to configure from AWS portal.



# Machine Learning Services

- IBM Watson, Cortana Intelligence, Google ML services, AWS ML
  - State of the art computer vision
  - Sophisticated text analysis
  - Automatics language translation
  - Tools to compose new ML tools
- The focus of cloud tech investment



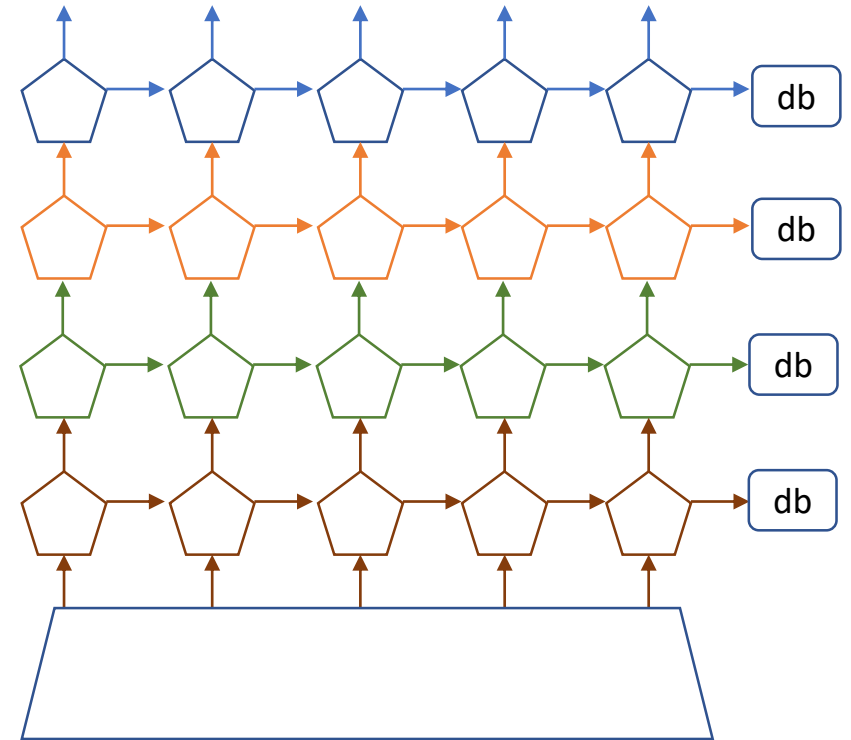
↑ Azure ML  
← Google Cloud ML

# How do you build these planet scale service?

- Requirements:
  - Global scale => distributed => well defined and usable consistency models
  - Dynamic scaling to support 1000s of concurrent peak time users.
  - You must assume the infrastructure is constantly failing
    - But your app must stay up!
  - Designed so that upgrade and test occur seamlessly while app is running.
  - Security and Privacy not afterthoughts.
- A style of application construction has evolved to support this.
  - ***Cloud Native***

# Cloud Native Applications

- Build app from the ground up from small, stateless “microservice” containers or functions
  - Supports scalable parallelism
  - Rapid application modification and evolution
  - Easily distributed to provide fault tolerance
- Examples:
  - Netflix, Facebook, Twitter, Google Docs
  - Azure CosmosDB – billions of transactions per week
  - Azure Event hub – trillions of requests per week
  - Azure Cortana – 500 million evals/sec
  - Azure IoT Hub, Skype, Power BI, CRM Dynamics
  - AWS Kinesis

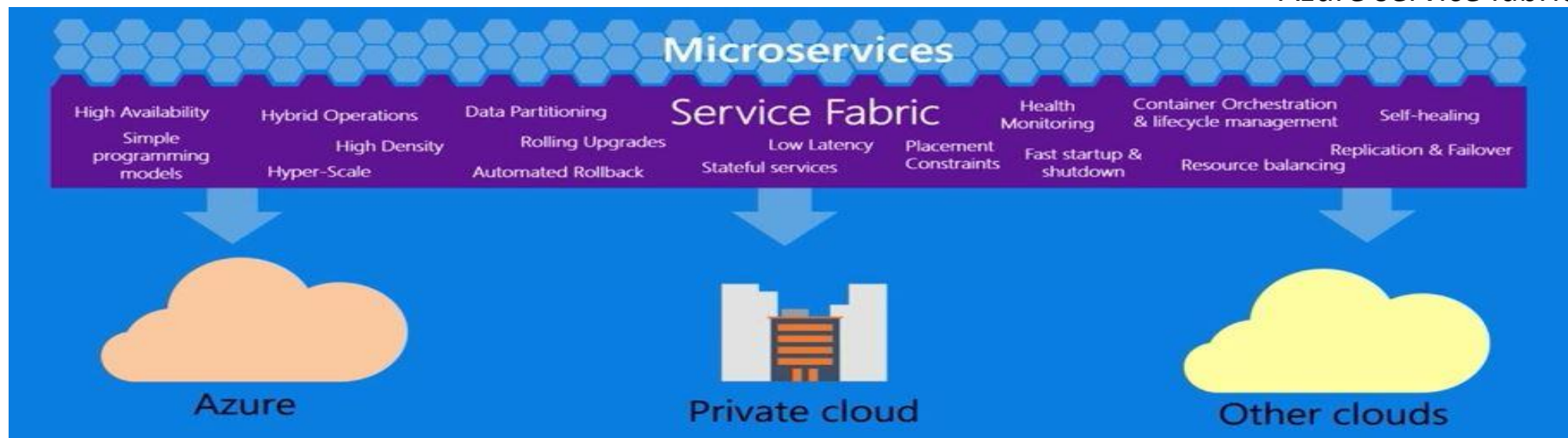




# How to Managing 1000s of Microservices?

- You need a service that can
  - Schedule containers across dozens of data centers
  - Handle fault monitoring and replication
  - Scale up and down when needed
  - Manage network proxies
- Kubernetes released by Google
  - Becoming a standard.
- Supported on Google, AWS, Azure, IBM, ....
- Mesos, Swarm are similar.

Azure service fabric

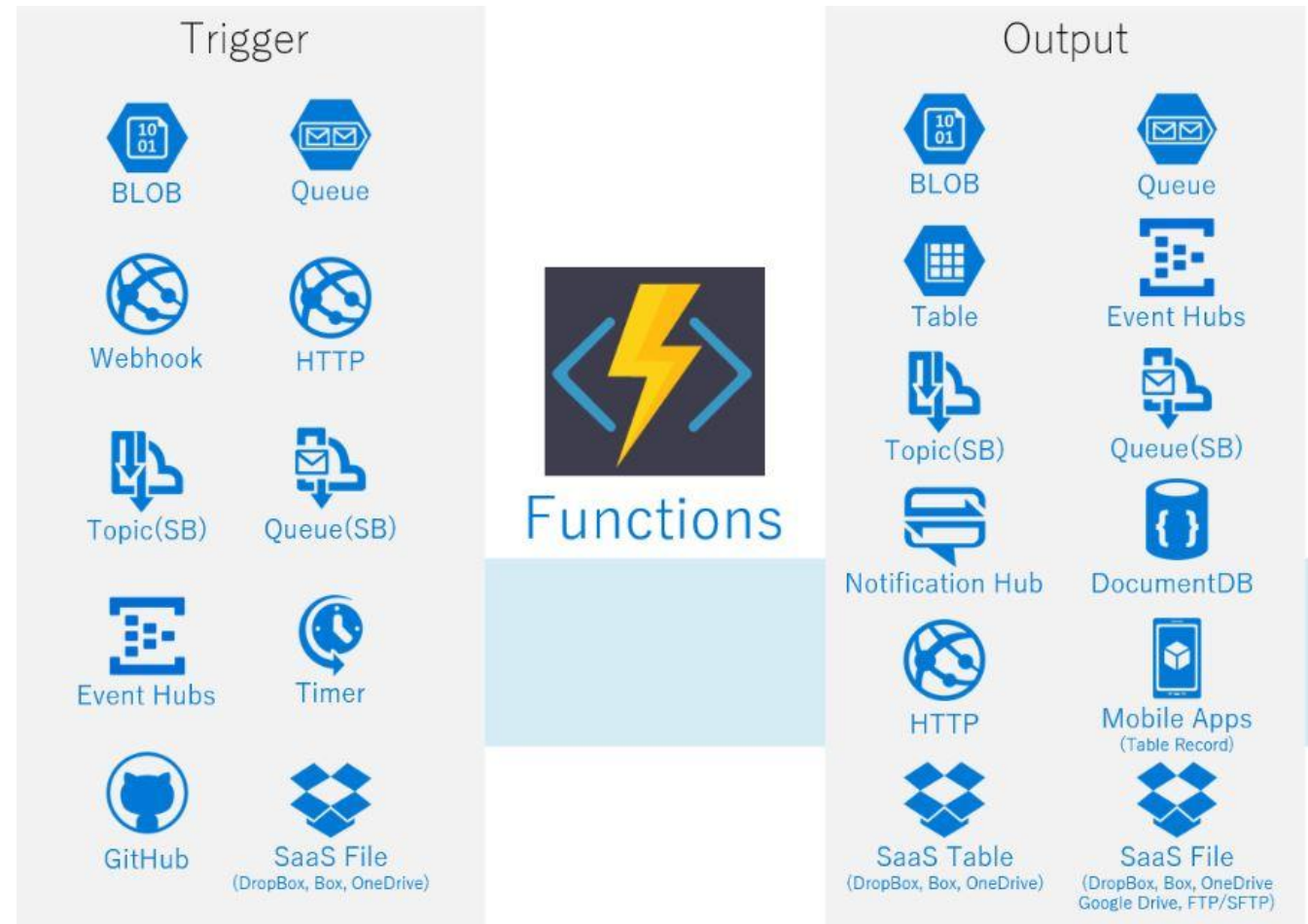


# Serverless Computing

- When I use a cloud service like CosmosDB or Kinesis do I need to allocate a server and deploy a virtual machine?
  - Of course not. Just configure it from the portal and use it.
- What if I have a simple function that I want to run every time I a particular “event” happens:
  - a file in the file service is modified
  - A specific external event is sent to a cloud event stream
  - Somebody added a file to a Github repository.
  - It is 2:00 pm.
- What are my choices?
  - Create a VM or container with my function and run it continuously.
  - But I don't want to pay for it when it is being idle.

# Serverless Functions as a Service

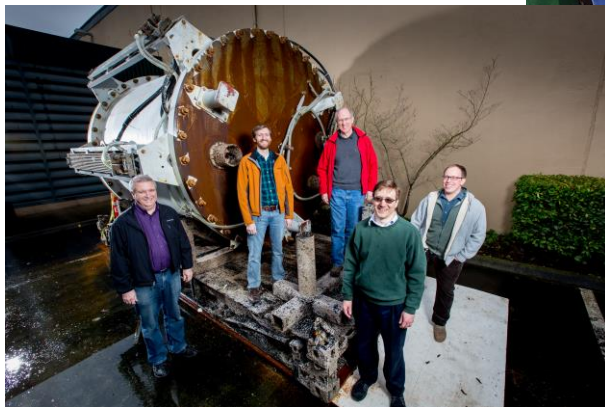
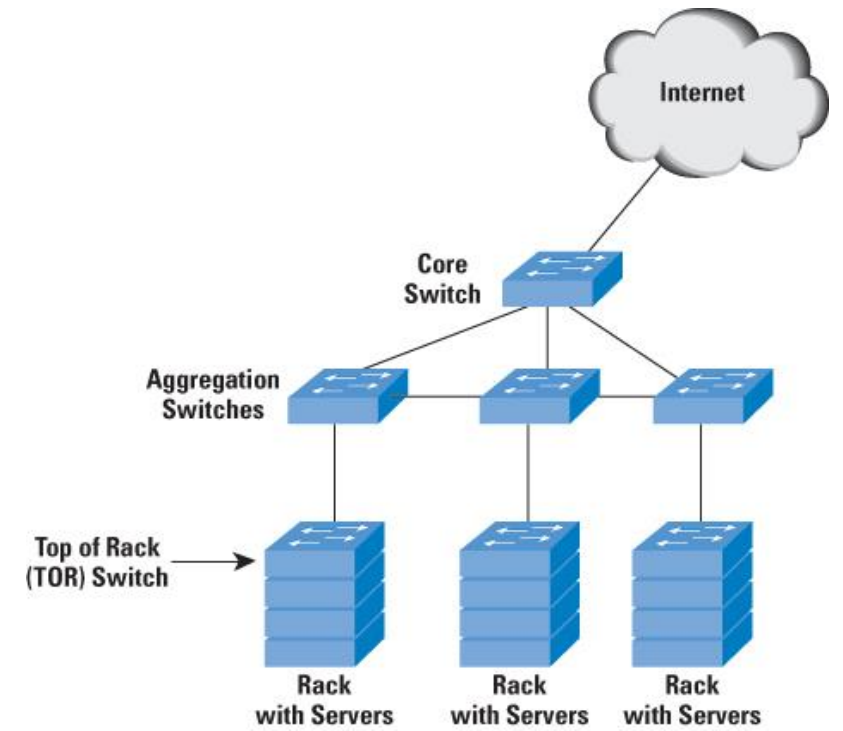
- AWS Lambda, Azure Functions, Google Functions, IBM OpenWhisk
- short-running, stateless computation
- driven by “triggers”
- scales up and down instantly and automatically
  - Can have hundreds of instances responding to events at once.
- based on charge-by-use
- Easy to configure from cloud portal



# The evolution of the data center

# Huge strides and experiments

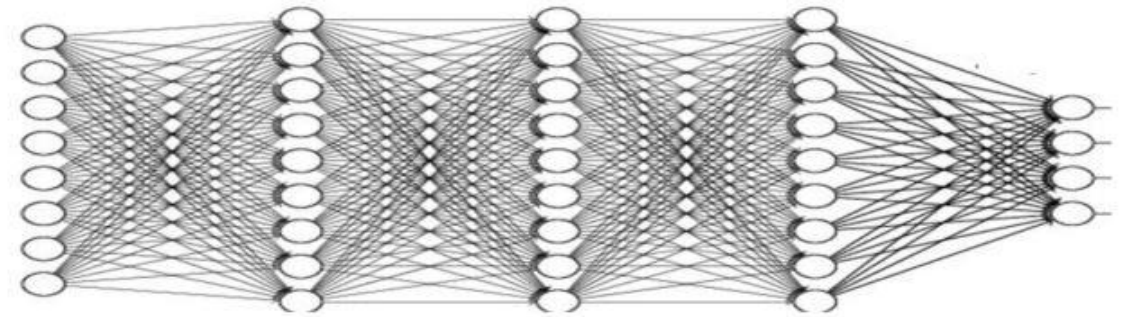
- Early days: 2005
  - Very simple servers
  - Network outward facing poor interconnect
- 2008-2016
  - **Software defined networks**
  - Special InfiniBand sub networks
  - Many different server types
    - 2 cores to 32 cores to GPU accelerations
  - Efficiency experiments
    - Geothermal, wind, wave
    - Containerized server





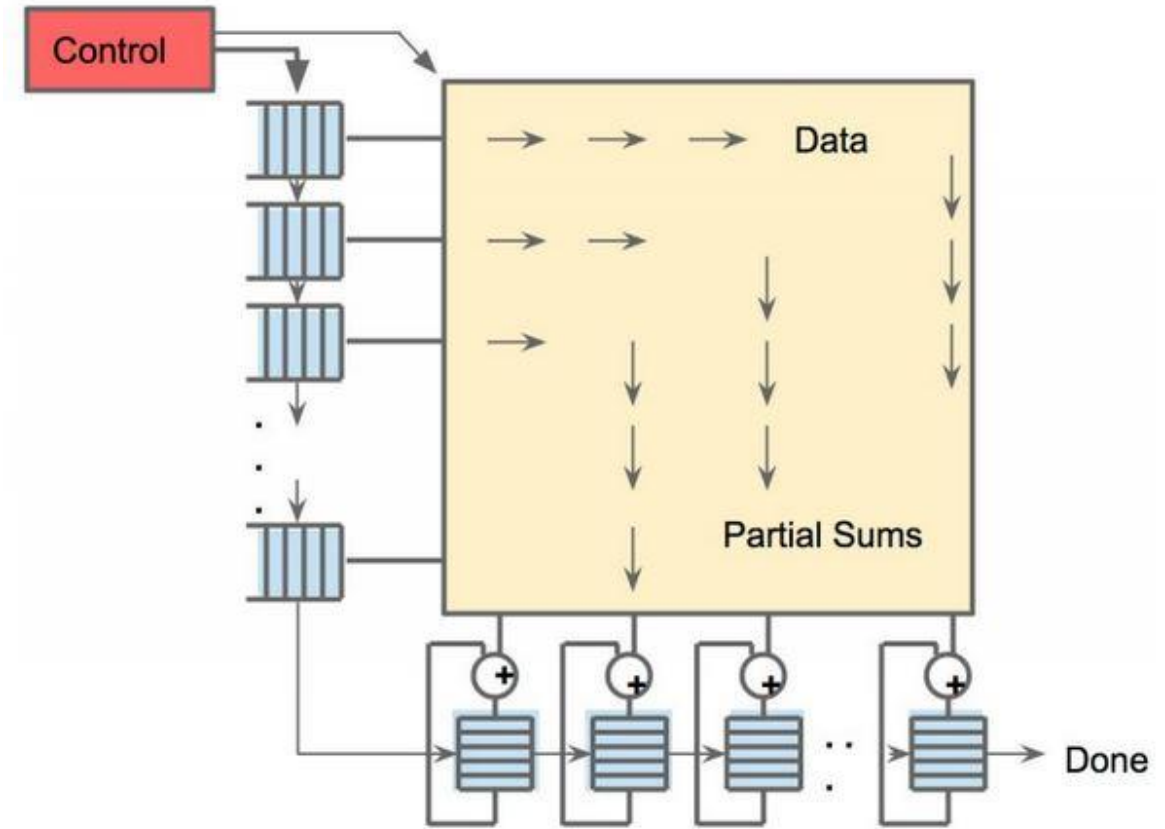
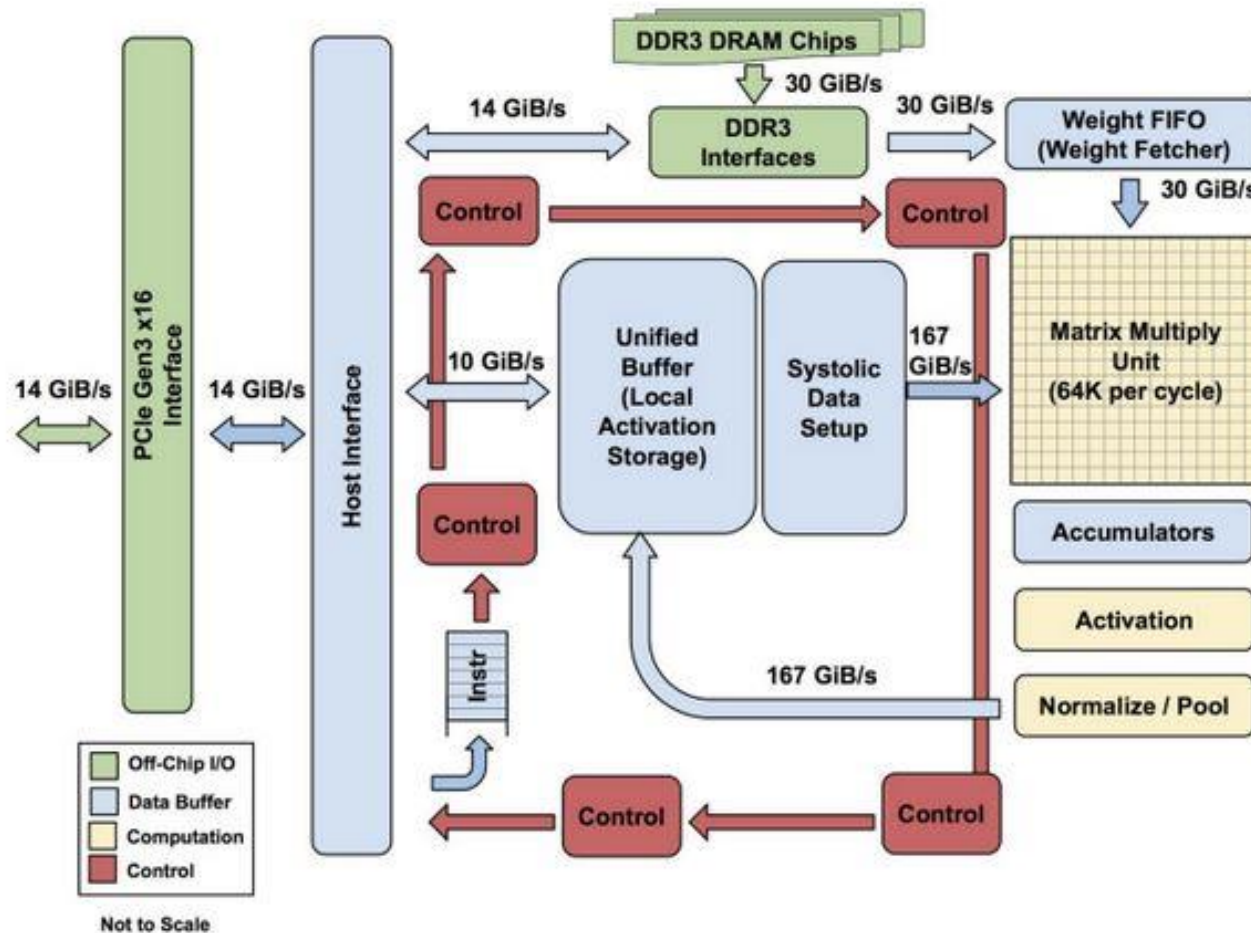
# Hardware direction

- Currently driven by advanced compute intensive on-line apps
  - Voice recognition and language translation
  - Image recognition
  - Search and analysis
- Machine learning a major driver
  - 2 phases: training and inference (prediction)
  - Specialized processor nodes are needed.



# Google Tensorflow chip

- optimize the response time of NN inference

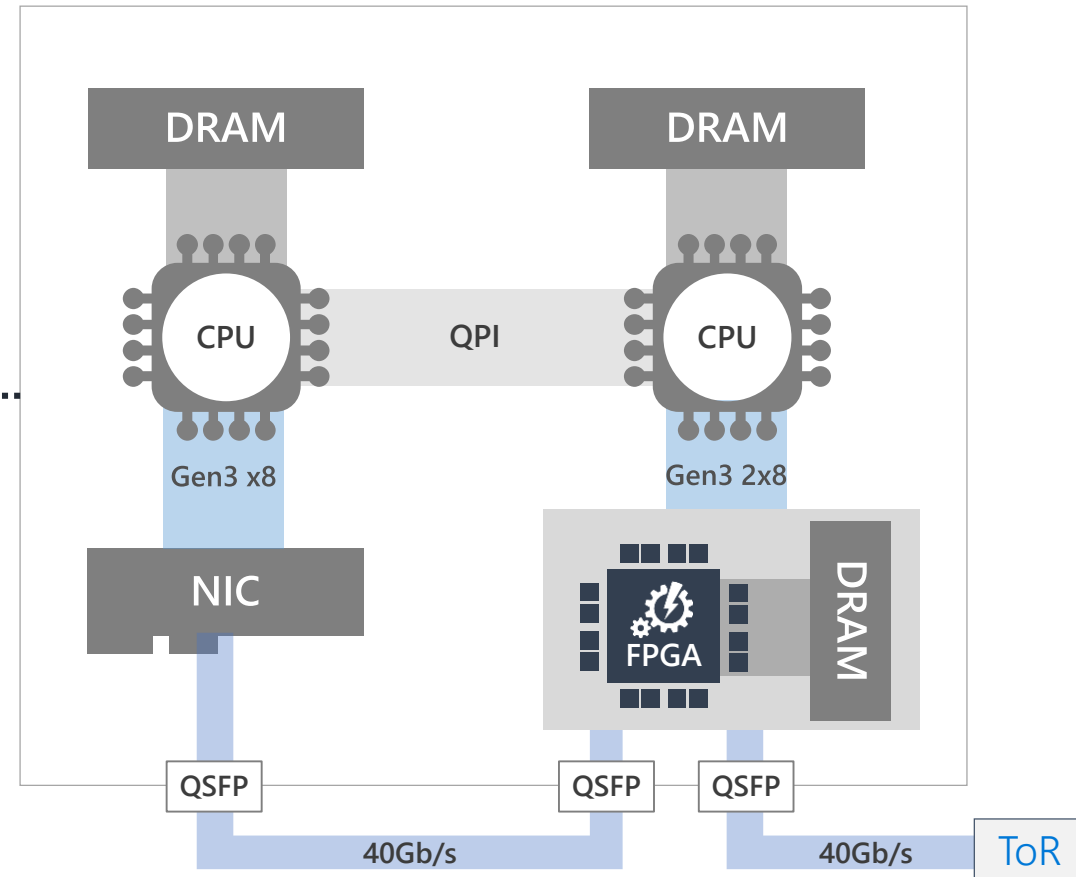
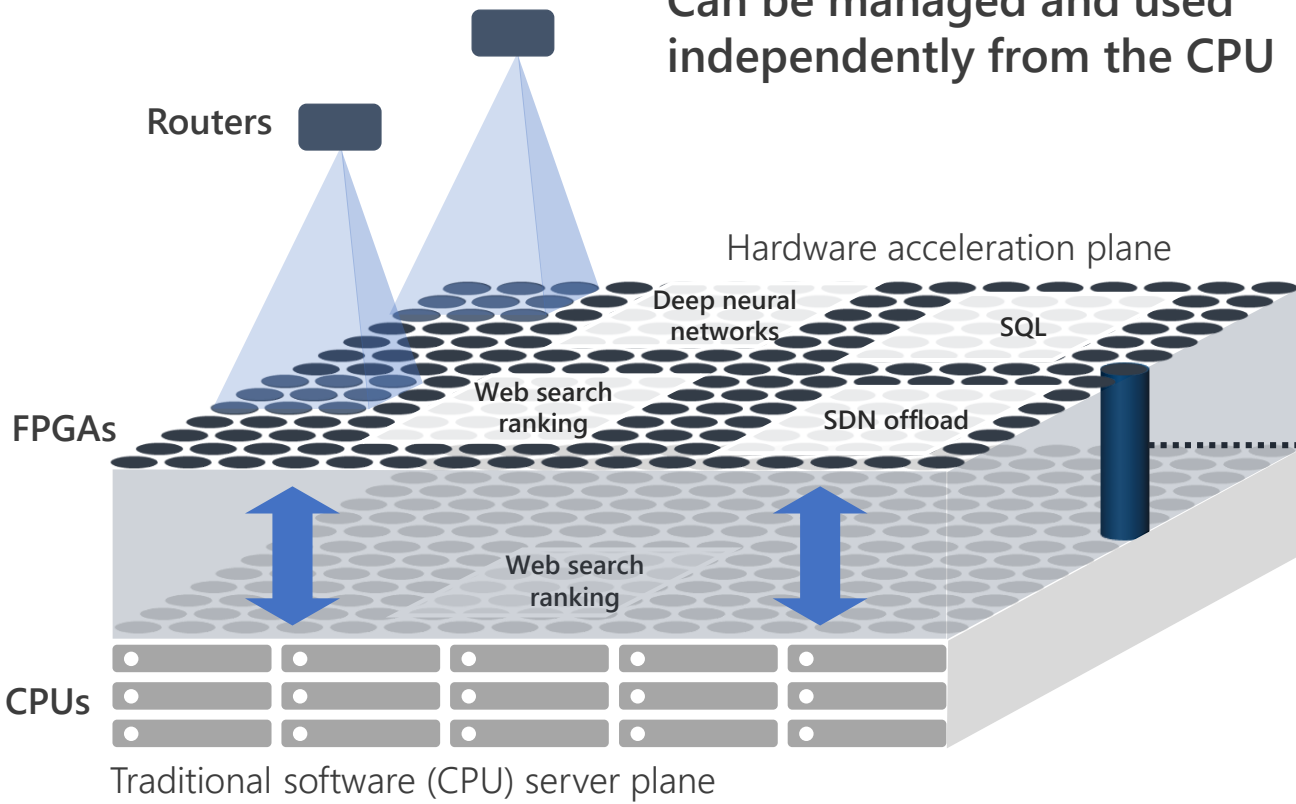


Systolic matrix multiply

# Hardware Microservices on FPGAs [MICRO'16]

Interconnected FPGAs form a separate plane of computation

Can be managed and used independently from the CPU



Where Things are Headed

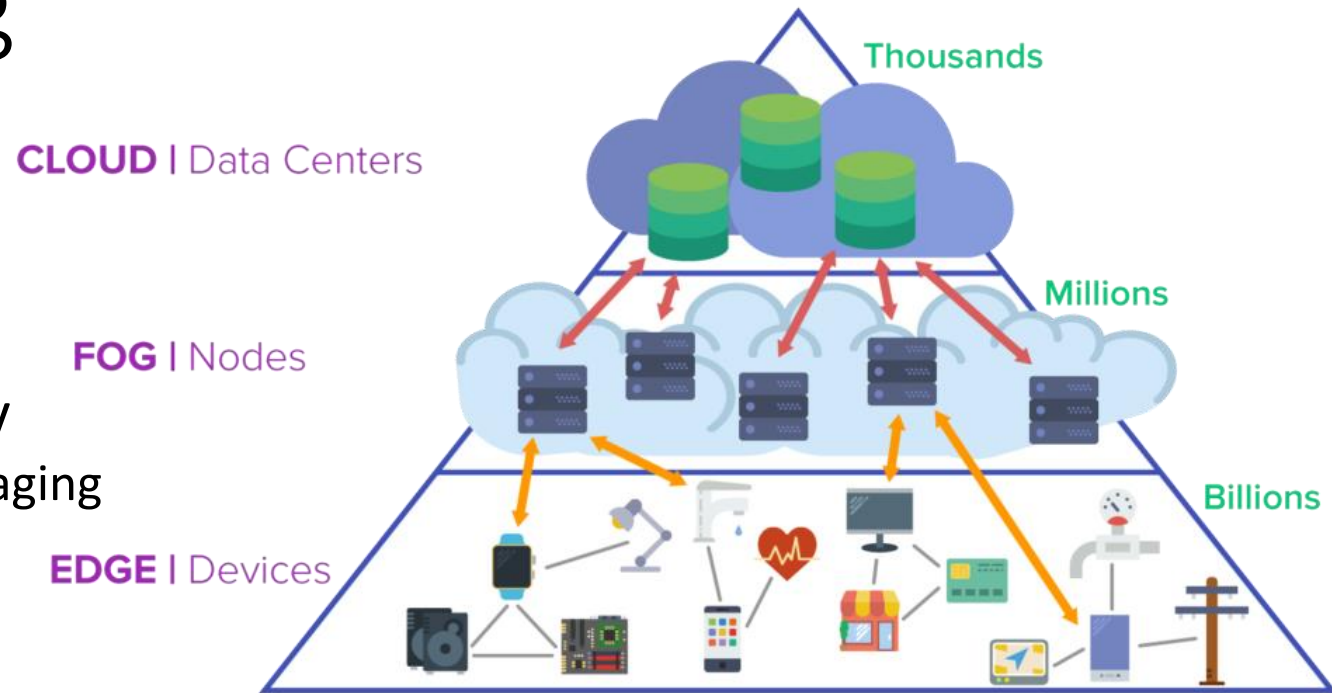
# Cloud as Supercomputer?

- Not really:
  - Cloud is optimized for fast response for *services* supporting many concurrent clients.
  - Supers are optimized for fast execution of *programs* on behalf of a small number of users.
- But there is some convergence:
  - Better cloud networks to improve bisection bandwidth and reduce latency
  - Addition of cloud GPUs and other special hardware
  - When supercomputers scale they will learn to understand fault tolerance.



# The Edge and the Fog

- We content distribution networks
  - Deliver cached content quickly
- But we need more than content
  - The edge contains compute capability
    - Call it the fog of small servers each managing hundreds of devices
  - Preprocess events from local devices and respond quickly if needed
- Can we push/migrate lambda functions or microservices from the data center to the edge?
- Can we extend the cloud fabric easily to the “fog” nodes?



# Machine Learning Services will Become Amazing

- The rise of the digital assistants represent convergence of current cloud research
  - Echo/Alexa, Siri, Ask Google, Cortana
  - Cloud ML has extended our senses, but not our ability to reason.
- But it is not AI.
  - ***“Deep Learning Isn't a Dangerous Magic Genie. It's Just Math”*** - Oren Etzioni



Yeast



Streptococcus



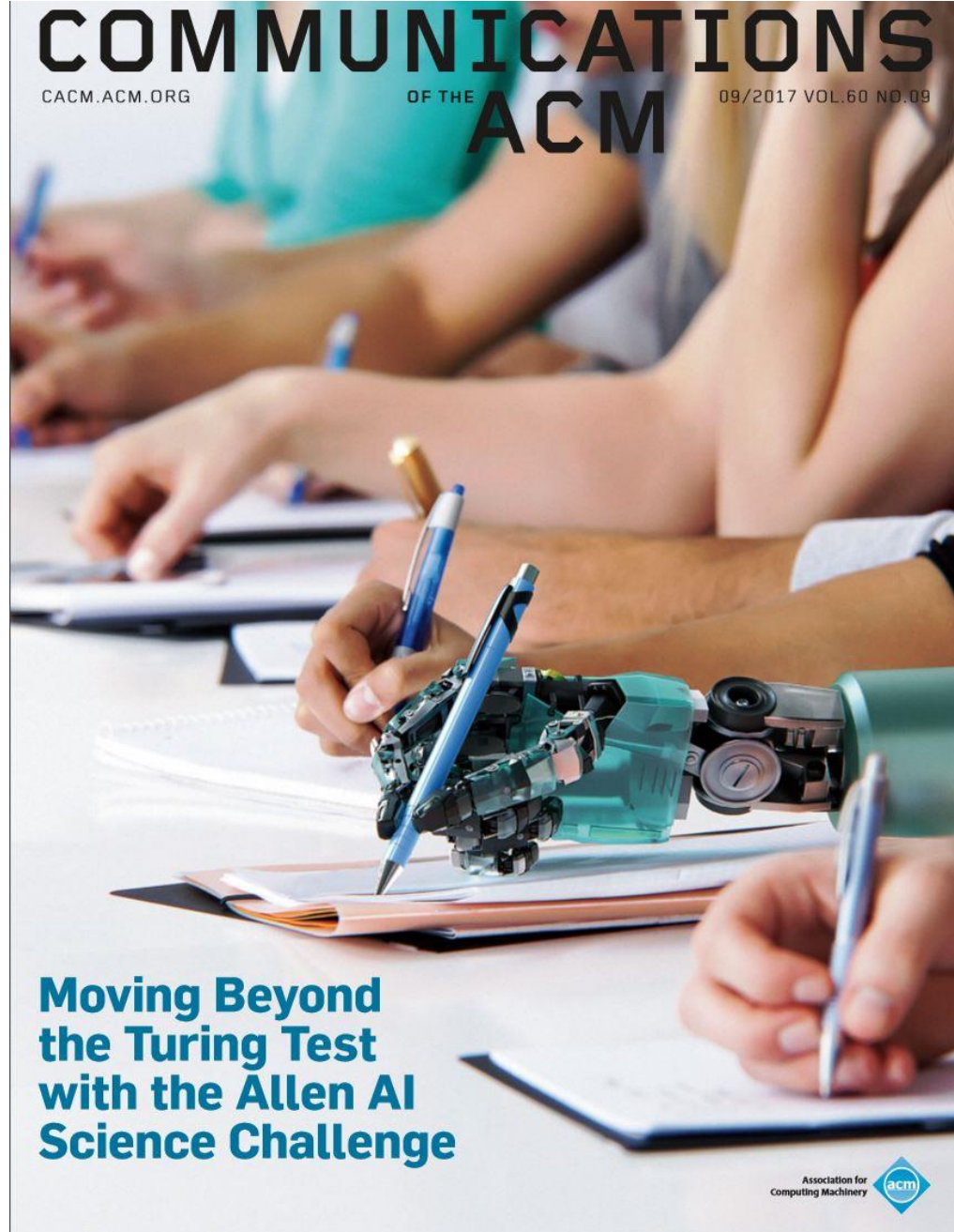
Amoeba



Seahorse

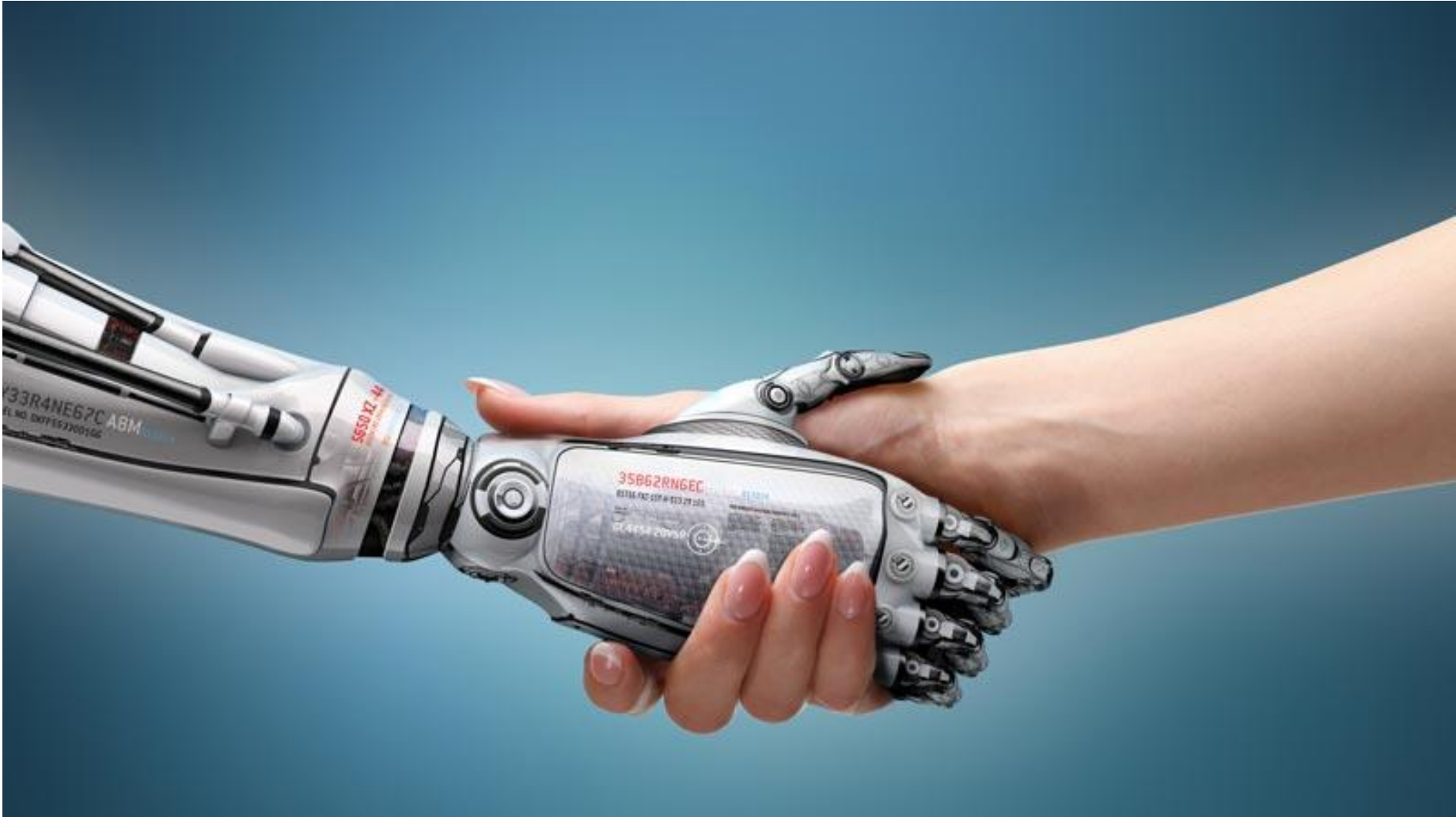
# Actual AI services?

- When can Alexa pass a 4<sup>th</sup> grade science exam?
  - Check out Aristo from Allen Institute for AI. They are getting close.
- The goal for Alexa, Siri, Google, Cortana:
  - A truly smart assistant.
    - *Siri: Please find the Higgs boson in this data.*
- Another goal: Driverless Car
  - Likely available everywhere by 2025
- What next?





# The Ultimate Edge of the Cloud



# Shameless Self-promotion

- The book “Cloud Computing for Science and Engineering”
- by Ian Foster and Dennis Gannon, published by MIT Press.
- Online here:
  - <https://Cloud4SciEng.org>

