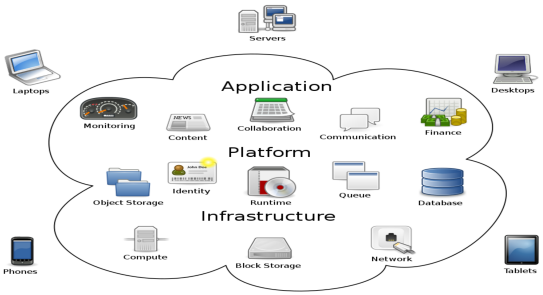
Meaning of life with Cloud Computing (and Azure)

What is Cloud Computing ?

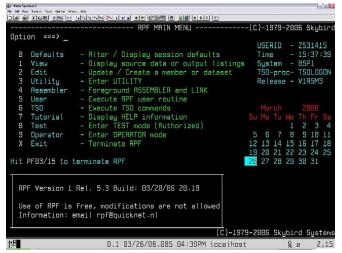


Cloud computing metaphor: networked elements providing services do not need not be addressed individually or be managed by users; instead, the entire provider-managed suite of hardware and software can be thought of as an amorphous cloud.
Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. Functionality is often distributed over multiple locations, each location being a data centre. Cloud computing relies heavily on sharing of resources.

Source: https://en.wikipedia.org/wiki/Cloud_computing

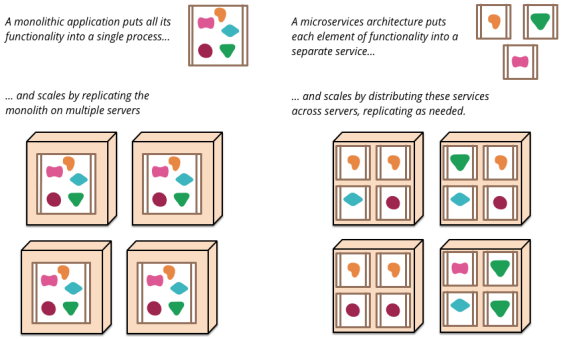
Is Cloud Computing something new?

Yes and no: On IBM's famous System/370 there was an OS called Multiple Virtual Storage (MVS) since the 1970ies. MVS was able to run multiple guest OS. All strictly decoupled from each other. In the beginning these where accessible with terminals over phone lines or dedicated RS232 lines. In the 1990ies IBM provided Internet connectivity. First via proprietary System Network Architecture (SNA). Later on via TCP/IP. Thus, the guest OS' were able to communicate even beyond data centre borders. Thereby offering a kind of basic strictly job based cloud with virtual OS instead of VM.
see https://en.wikipedia.org/wiki/MVS

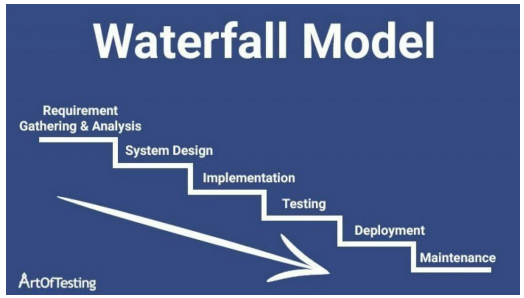


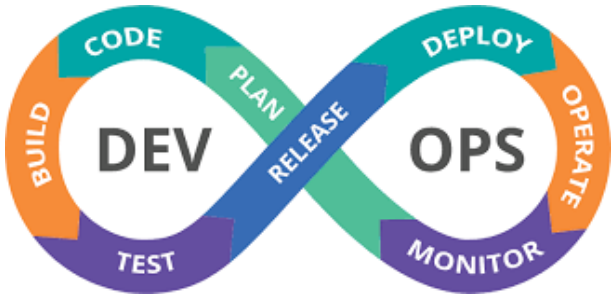
Cloud Computing encourages modern IT processes

To replace Monolith with Microservices

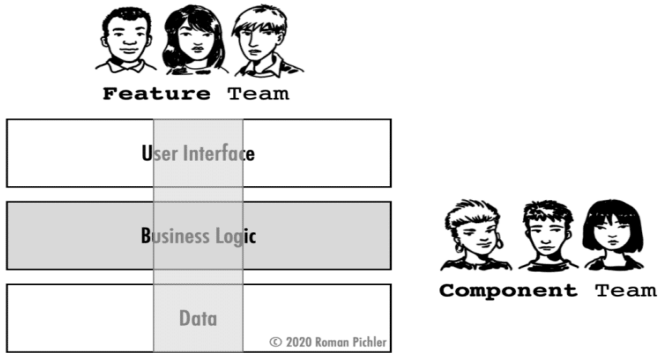


To replace Waterfall by iterative, agile DevOps





To replace component teams with feature teams



Why Cloud Computing ?

(from https://www.ibm.com/cloud/learn/benefits-of-cloud-computing)

Flexibility

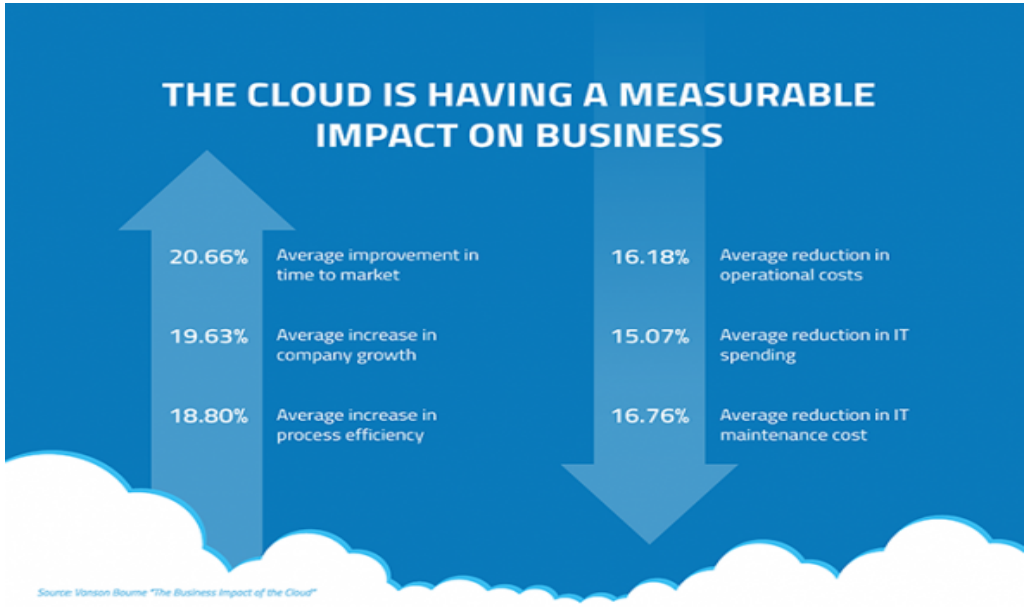
Users can scale services to fit their needs, customize applications and access cloud services from anywhere with an internet connection.

Efficiency

Enterprise users can get applications to market quickly, without worrying about underlying infrastructure costs or maintenance.

Strategic value

Cloud services give enterprises a competitive advantage by providing the most innovative technology available.



Flexibility

- **Scalability:** Cloud infrastructure scales on demand to support fluctuating workloads.

- **Storage options:** Users can choose public, private, or hybrid storage offerings, depending on security needs and other considerations.
- **Control choices:** Organizations can determine their level of control with as-a-service options. These include software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).
- **Tool selection:** Users can select from a menu of prebuilt tools and features to build a solution that fits their specific needs.
- **Security features:** Virtual private cloud, encryption, and API keys help keep data secure.

Efficiency

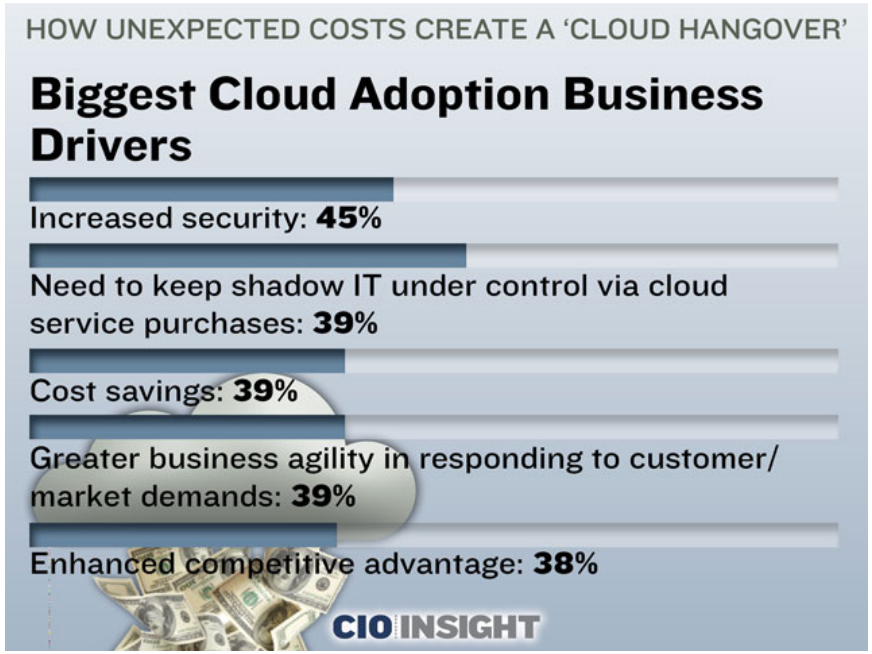
- **Accessibility:** Cloud-based applications and data are accessible from virtually any internet-connected device.
- **Speed to market:** Developing in the cloud enables users to get their applications to market quickly.
- **Data security:** Hardware failures do not result in data loss because of networked backups.
- **Savings on equipment:** Cloud computing uses remote resources, saving organizations the cost of servers and other equipment.
- **Pay structure:** A “utility” pay structure means users only pay for the resources they use.

Strategic value

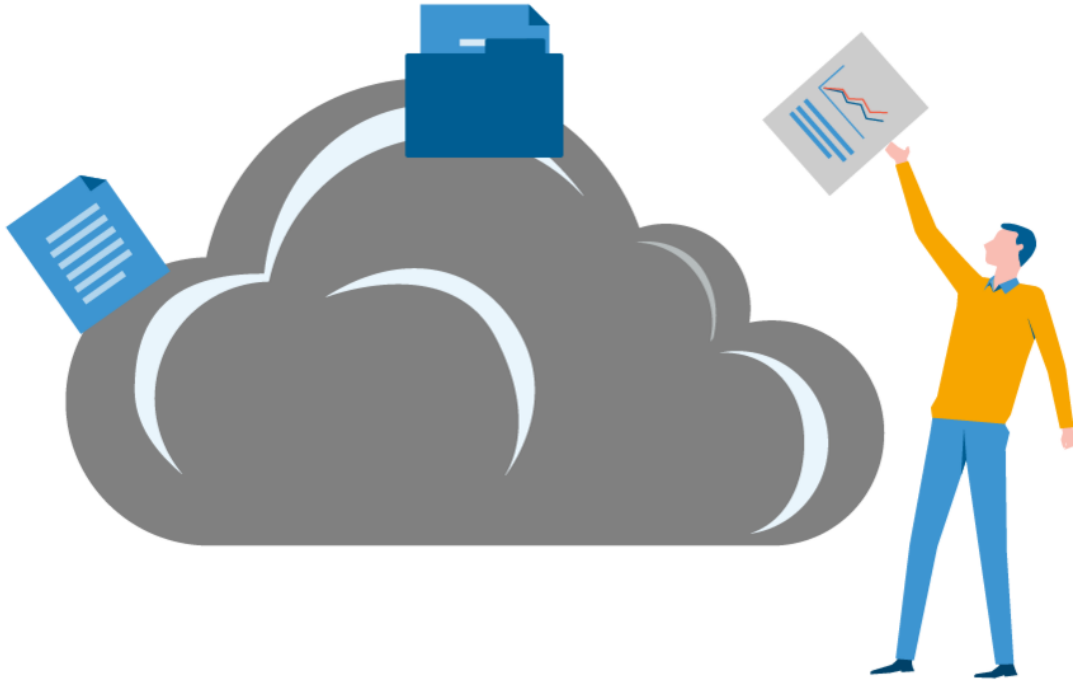
- **Streamlined work:** Cloud service providers (CSPs) manage underlying infrastructure, enabling organizations to focus on application development and other priorities.
- **Regular updates:** Service providers regularly update offerings to give users the most up-to-date technology.
- **Collaboration:** Worldwide access means teams can collaborate from widespread locations.
- **Competitive edge:** Organizations can move more nimbly than competitors who must devote IT resources to managing infrastructure.

Why not Cloud Computing

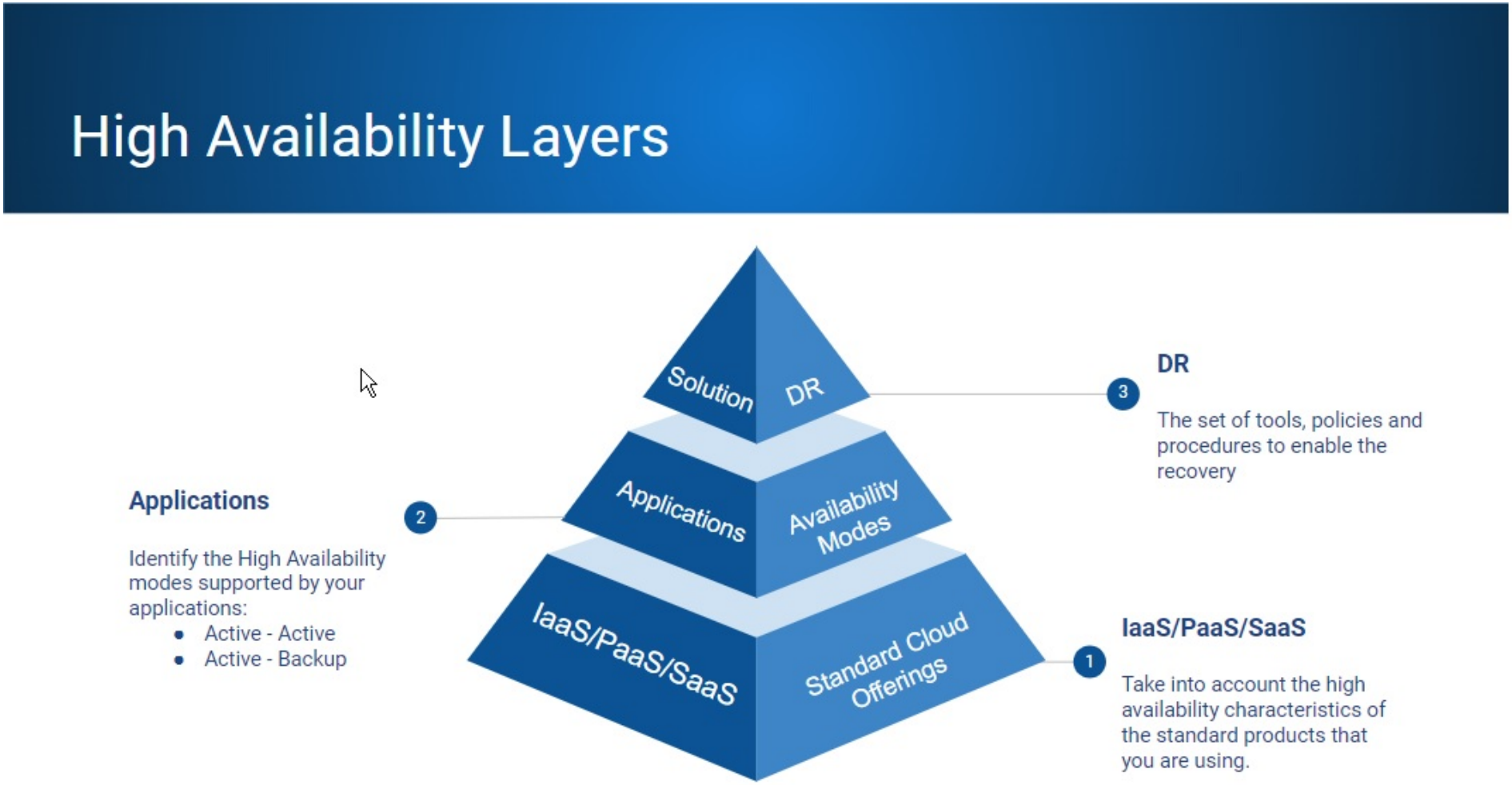
- Unexpected costs can threaten a company’s survival



- Security is not transparent

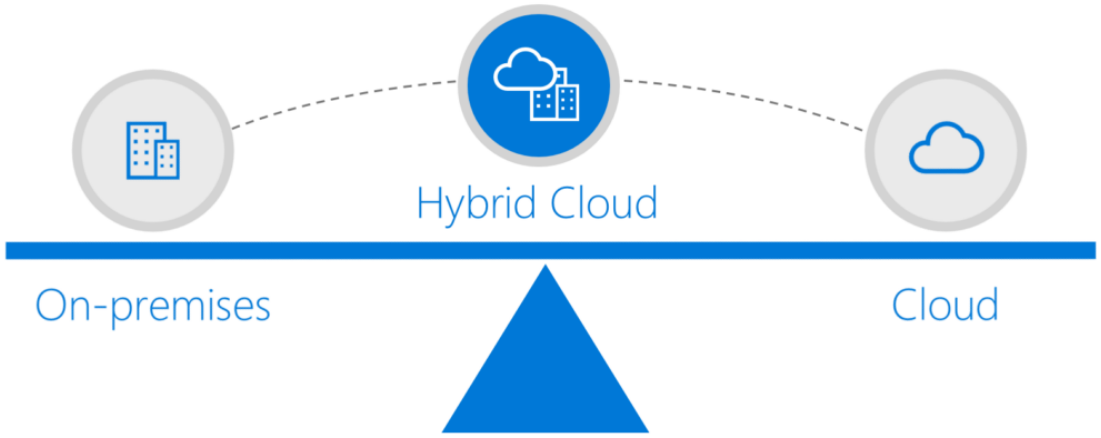


- Availability is not measured according to applications but services instead



Why Azure ?

Though Azure is far from being more advanced, more simple to use, or with lower pricing than other Cloud solutions it has one great advantage: you can migrate your existing on-premises (on-prem) Windows Server based structure nearly 1:1 to Azure. Furthermore, Azure offers great connectivity between cloud and on-prem. Therefore, most local data centres aim for a hybrid solution.



How to approach Azure ?

Very careful and in English

There are some great resources for learning Azure:

- An Azure student or test subscription under https://portal.azure.com/#home
- Microsoft learning paths at https://docs.microsoft.com/en-us/learn/azure/
- Github labs https://github.com/MicrosoftLearning

But all resources have one thing in common: there are either no translations or translations are quite bad and rudimentary. In addition all certifications are available in English only. As long as you don't speak Chinese, Japanese, or some such.
If your English is not sufficient, try https://www.deepl.com/en/translator.

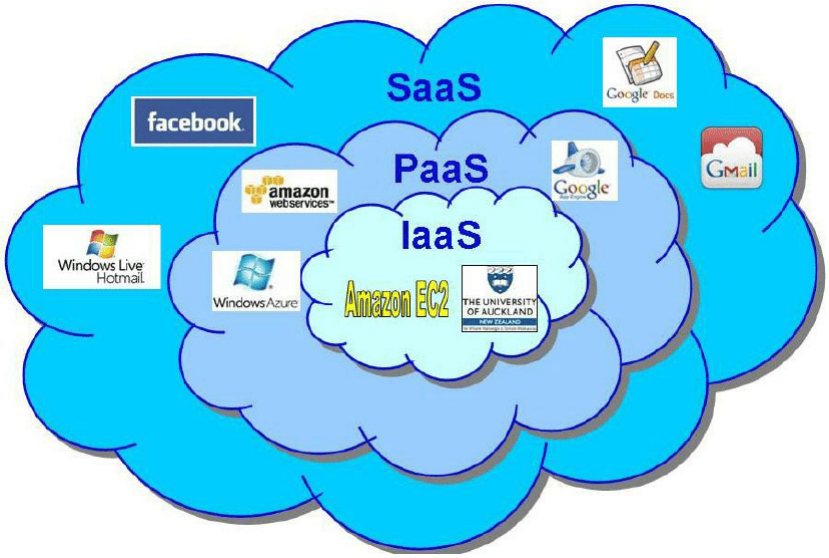
Pitfalls on our approach to Azure

- Inconsistent updates: Azure changes all the time without announcement. Web-GUI, learning paths, and Github labs do not change consistently with it. Thus, exercises which worked yesterday do not work today any more.
- Different capabilities of sponsored subscriptions: For example student account can use 4 CPUs max. in parallel. Test and teacher accounts can use up to 6 CPUs. It is unclear how many CPUs free sandboxes can use. Therefore, exercises tested by test accounts may not work with student accounts. Numbers are of this writing and may change without announcement.
- Confusing acronyms and terms: Microsoft sometimes redefines well known acronyms and terms.
  see https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-abbreviations
- Different implementation of commonly used on-prem services: most important example is Active Directory (AD). The main administrative tools in Windows Server's AD of Group Policy Objects (GPO) and Organisational Unit (OU) are **not** available in Azure's AD.

Basic concepts of Cloud Computing

Cloud computing Platform Services

Amazon’s web shops are missing from this picture. They are SaaS.



SaaS – Software as a Service

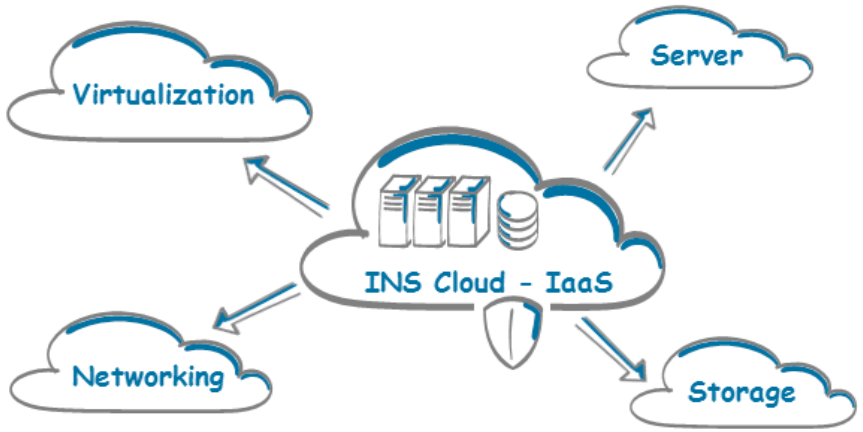
Software as a service is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. SaaS is also known as "on-demand software" and Web-based/Web-hosted software. Source: https://en.wikipedia.org/wiki/Software_as_a_service

Example: Office365

IaaS – Infrastructure as a Service

Infrastructure as a service are online services that provide high-level APIs used to dereference various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc.
Source: https://en.wikipedia.org/wiki/Infrastructure_as_a_service

Examples: Virtual networking, virtual storage, serverless computing.



Infrastructure as a Service

- Server
- Storage
- Networks
- Low total costs (TCO), investment protection
- Rapid deployment
- Scalable upon demand
- Service Level Agreement (SLA), guaranteed availability, time of reaction and solution
- Hosted in high available TIER3+ data centers located in Germany

PaaS – Platform as a Service

Platform as a service (PaaS) or application platform as a service (aPaaS) or platform-based service is a category of cloud computing services that allows customers to provision, instantiate, run, and manage a modular bundle comprising a computing platform and one or more applications, without the complexity of building and maintaining the infrastructure typically associated with developing and launching the application(s); and to allow developers to create, develop, and package such software bundles.

Examples: Azure Search, Azure Content Delivery Network (CDN).

Azure Infrastructure

Azure Virtual Networking and Azure Express Route (not covered further here)

See https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview

See https://docs.microsoft.com/de-de/azure/expressroute/expressroute-introduction

Features

- Virtual Networks with Subnets
- Virtual switches and routing
- Peering between virtual networks
- Firewalls for IP-traffic and content based application firewalls
- Load balancing
- Gateways for applications and services, VPN-gateways

Azure Virtual Machines (not covered further here)

See https://docs.microsoft.com/en-us/azure/virtual-machines/windows/overview

Features

- All types of OS possible
- Essentially identical to Hyper-V-VMs, even with nested hosting

Azure Virtual Storage (not covered further here)

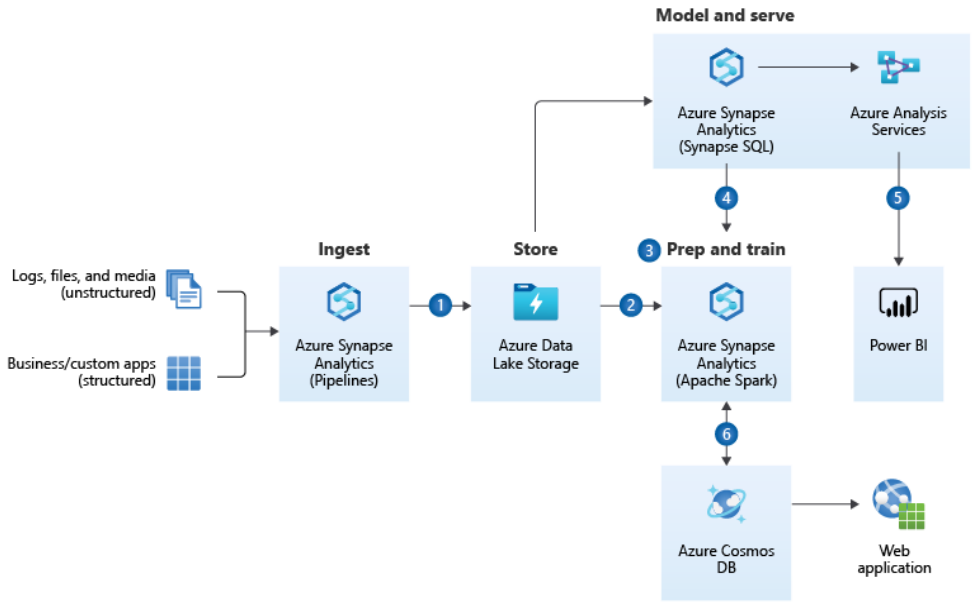
See https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction

Features

- Consists of: Block Blob storage, page Blob storage, append Blob storage, table storage, file storage , queue storage, Azure (managed) disks
- Different storage tiers: archive, cool, hot with different performance attributes and different pricing
- Different performance tiers: basic, standard, premium

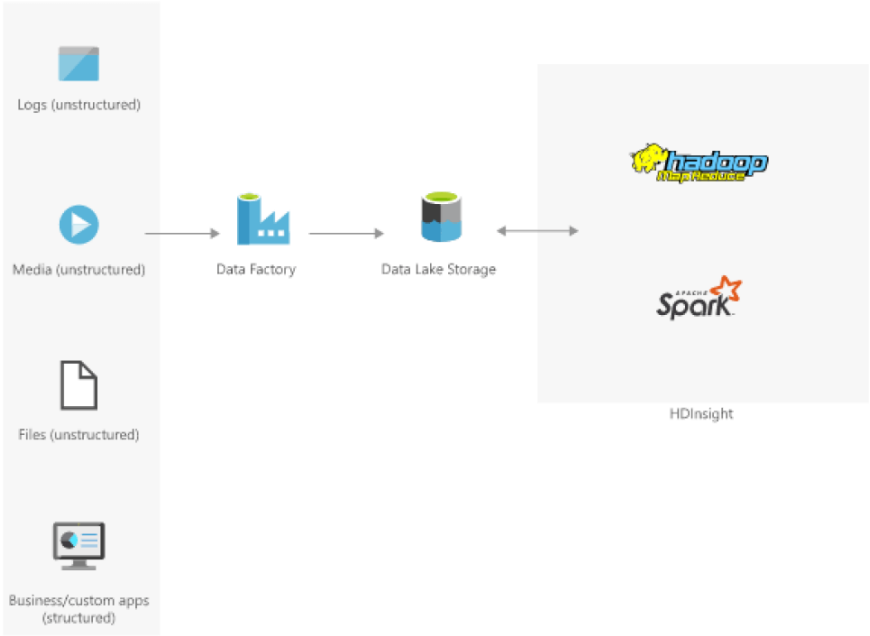
Azure Big Data (not covered further here)

See https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction



Features

- Azure Ananlytics
- Azure Data Lake
- Azure Cosmos DB
- Azure Hindsight



Features of Hindsight

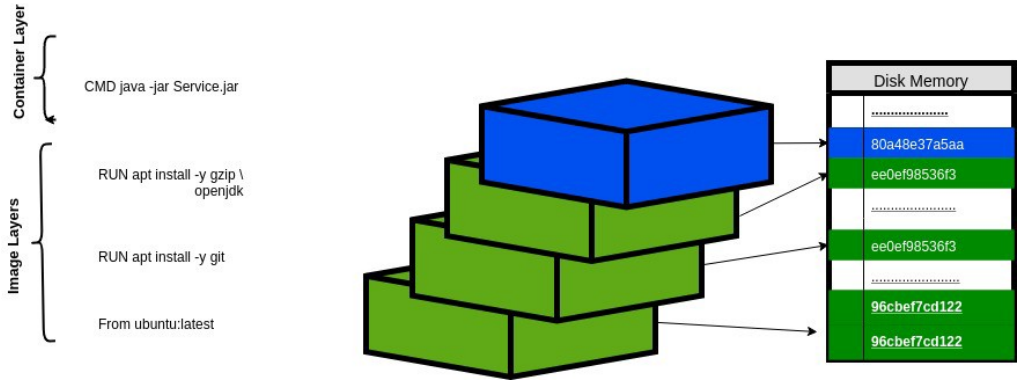
- Apache's Spark (https://en.wikipedia.org/wiki/Apache_Spark)
- Apache's hadoop (https://en.wikipedia.org/wiki/Apache_Hadoop)
- Usually connected to Azure Data Lake Storage

Azure Container

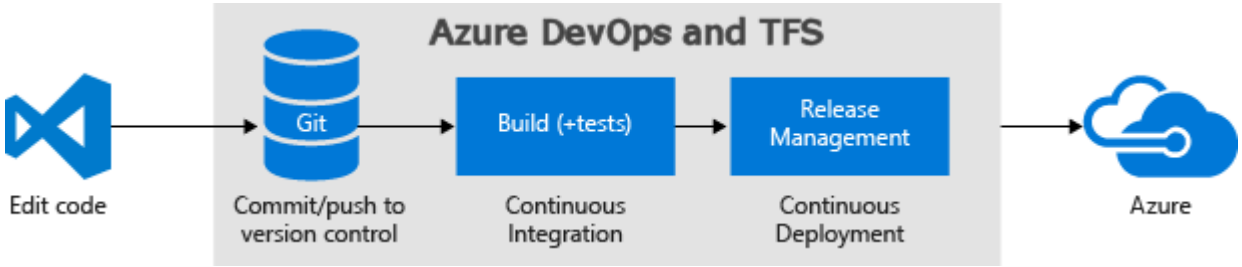
Features

- Docker based containers for arbitrary OS and applications

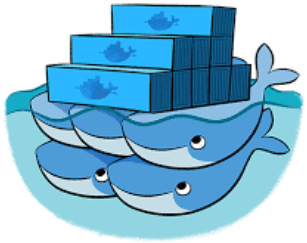
UnionFS for containers



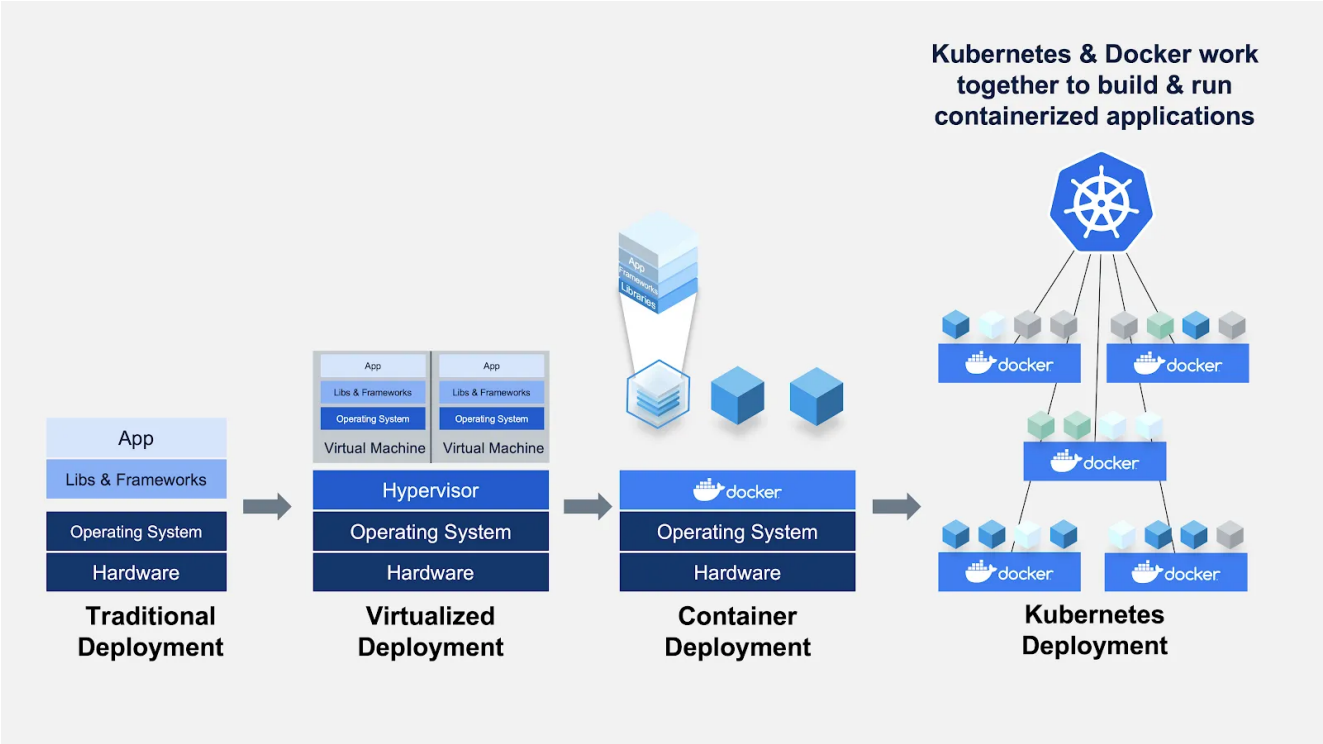
- DevOps with Team Foundation Server (TFS)



- Orchestration with Docker’s Swarm (very rare) or Kubernetes (de facto standard)



Docker Swarm

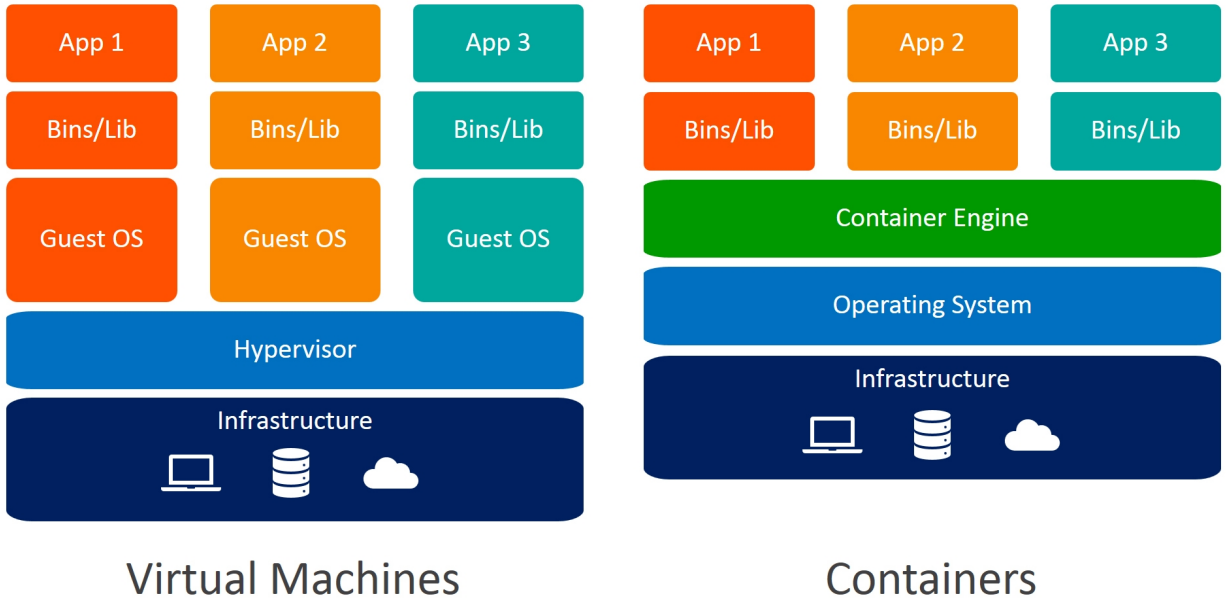


Evolution of Deployment

Windows/Azure Container Caveat

While Azure uses normal containers for Linux et al. it cannot do so for Windows based applications due to bad system architecture. Instead it uses a VM, albeit reduced to barest minimum: a so called Nano server which is essentially a headless Windows Server OS. Its size revolves at about 512 MiB. For this abomination Microsoft coined the term “Hyper-V isolation” as opposed to a normal container which does “application isolation”. This results in the fact that most of the advantages of a Container do not exist for Windows applications.

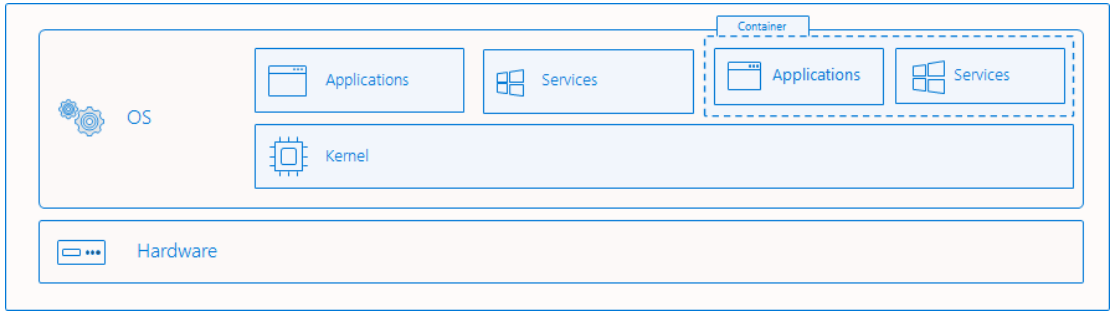
Remember the difference between VM and Container:



In contrast see the following page from https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/hyperv-container This page tries to sell a critical architectural flaw as an advantage.

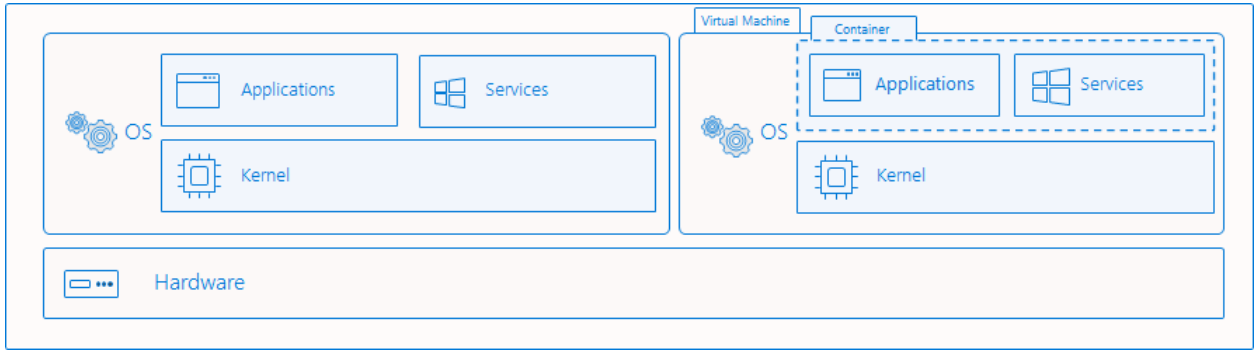
Process Isolation

This is the "traditional" isolation mode for containers and is what is described in the Windows containers overview. With process isolation, multiple container instances run concurrently on a given host with isolation provided through namespace, resource control, and process isolation technologies. When running in this mode, containers share the same kernel with the host as well as each other. This is approximately the same as how Linux containers run.



Hyper-V Isolation

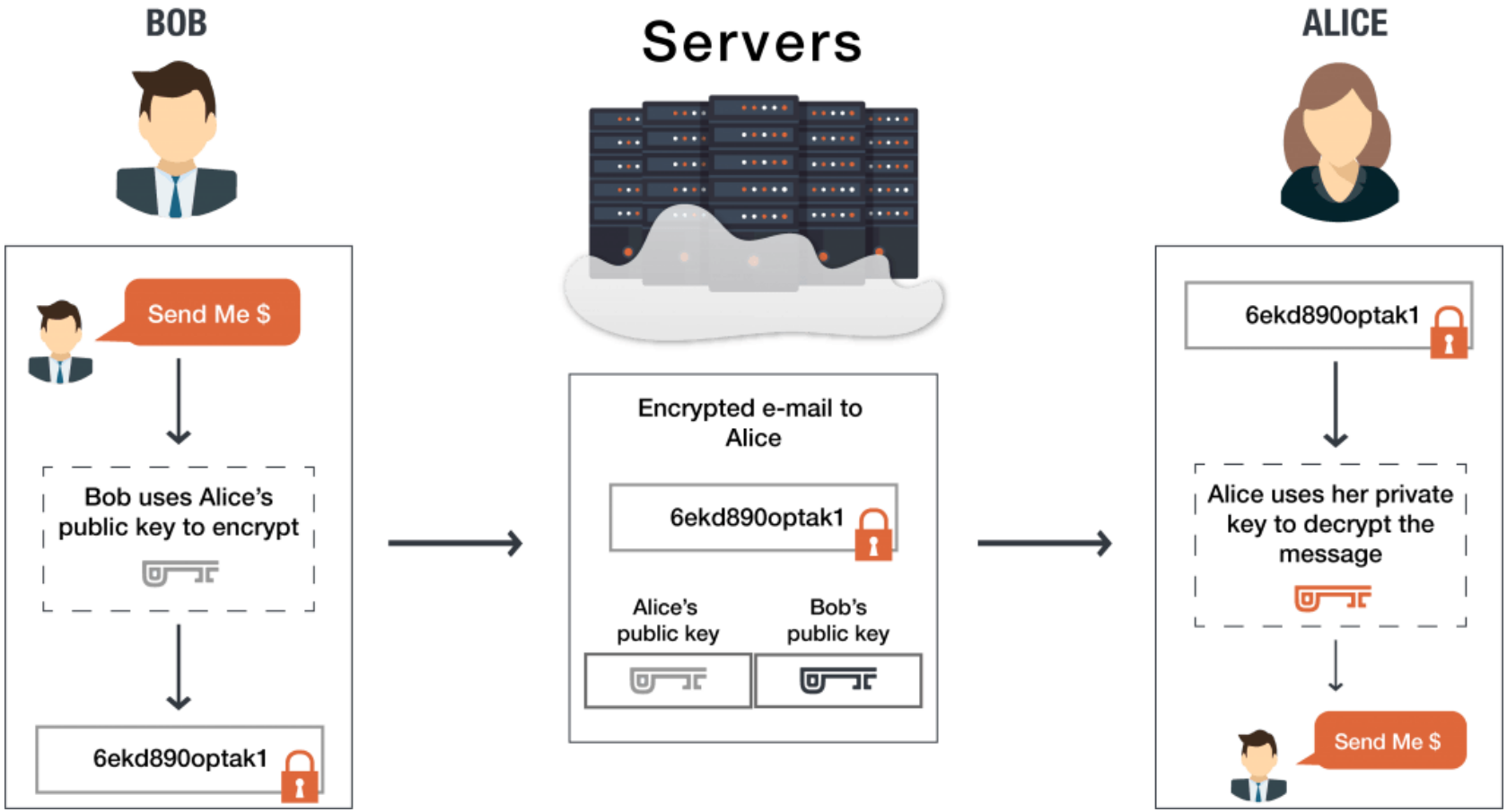
This isolation mode offers enhanced security and broader compatibility between host and container versions. With Hyper-V isolation, multiple container instances run concurrently on a host; however, each container runs inside of a highly optimized virtual machine and effectively gets its own kernel. The presence of the virtual machine provides hardware-level isolation between each container as well as the container host.



Azure Cloud Security

European Union’s data protection laws demand end to end encryption

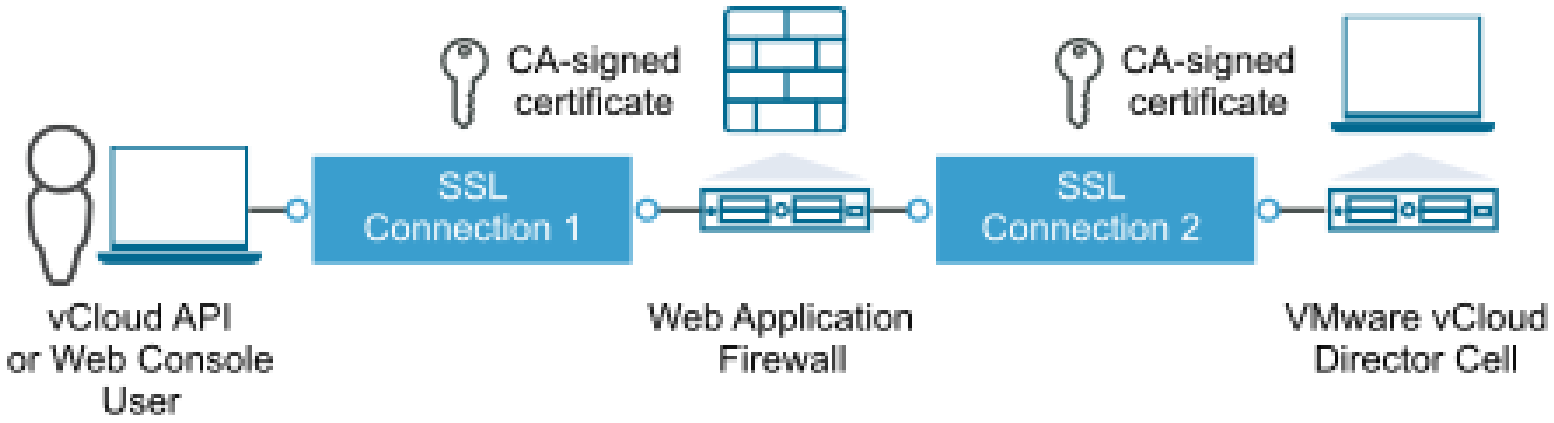
A penalty of up to **4% of yearly gross turnover for the whole company** is pending



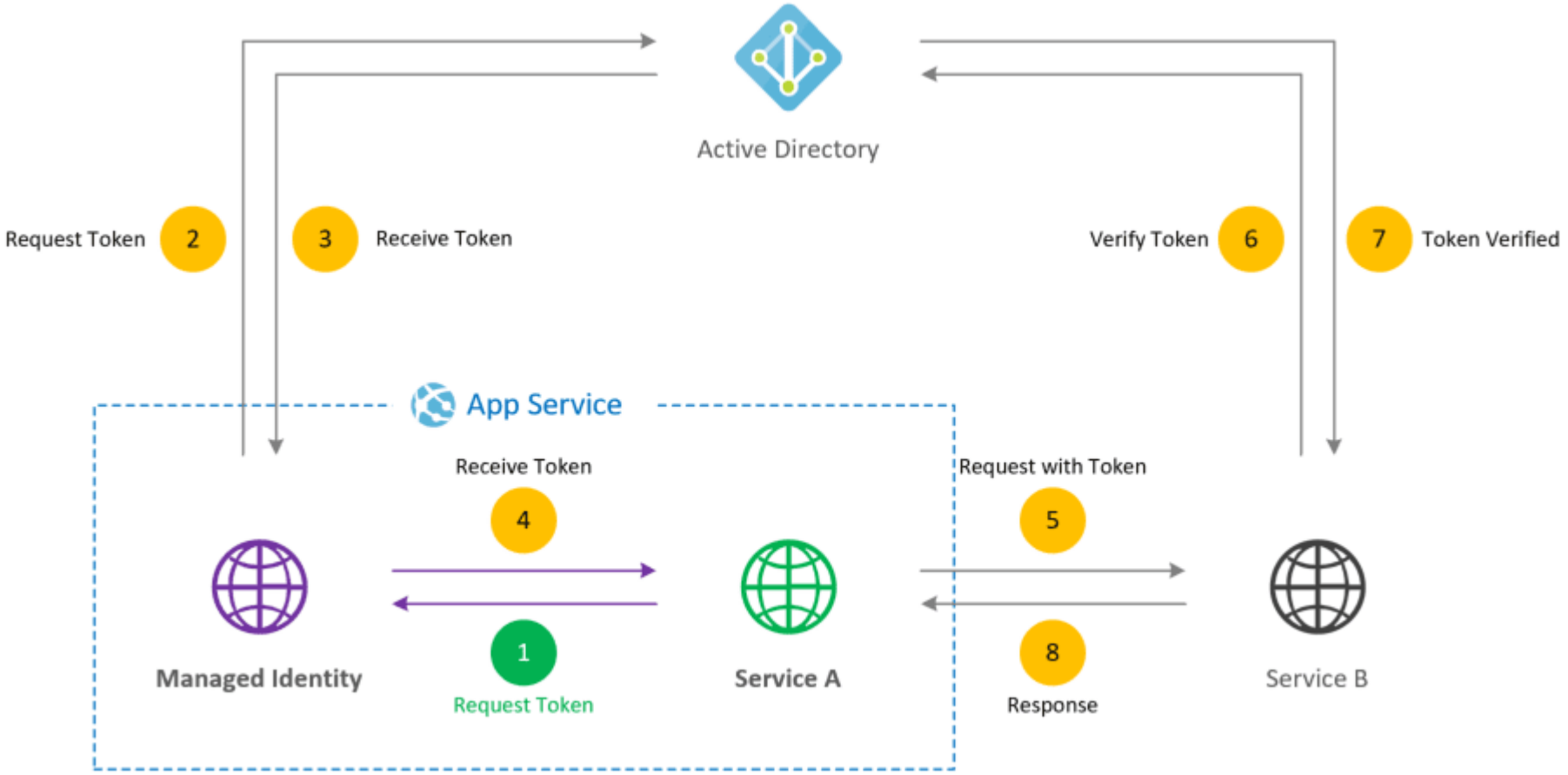
Reality with Azure and most other Clouds except Red Hat: Encryption offloading

(Mostly with security as a service IaaS)

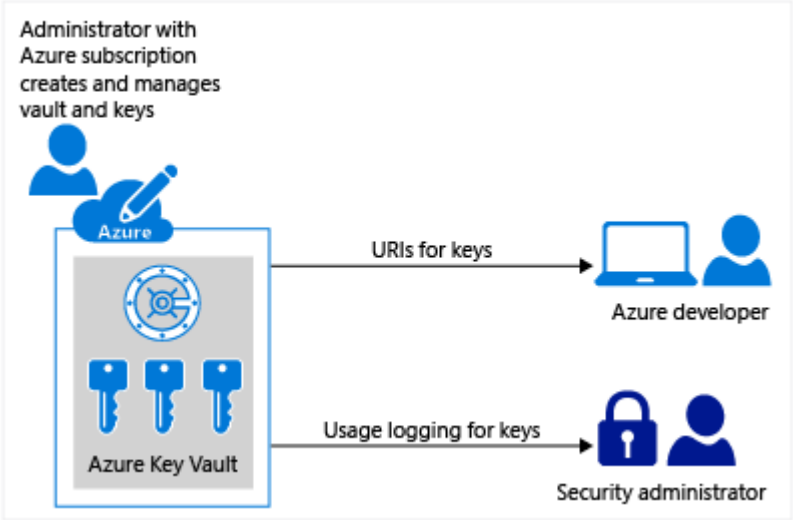
DO NOT DO EITHER! Neither Security as a Service nor encryption offloading.

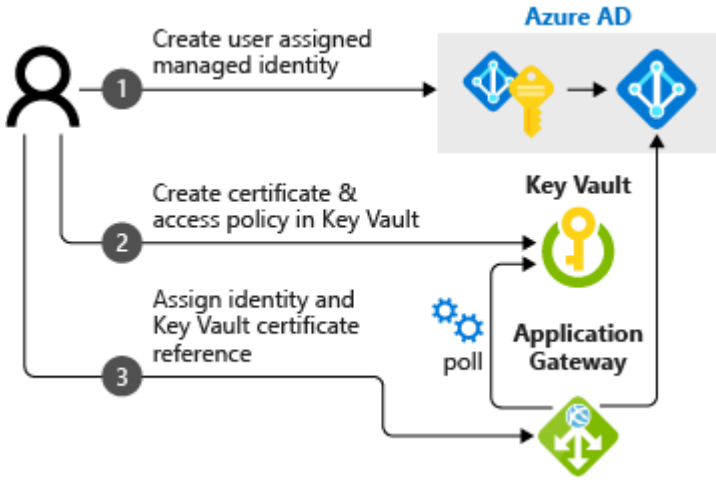


- Managed identities

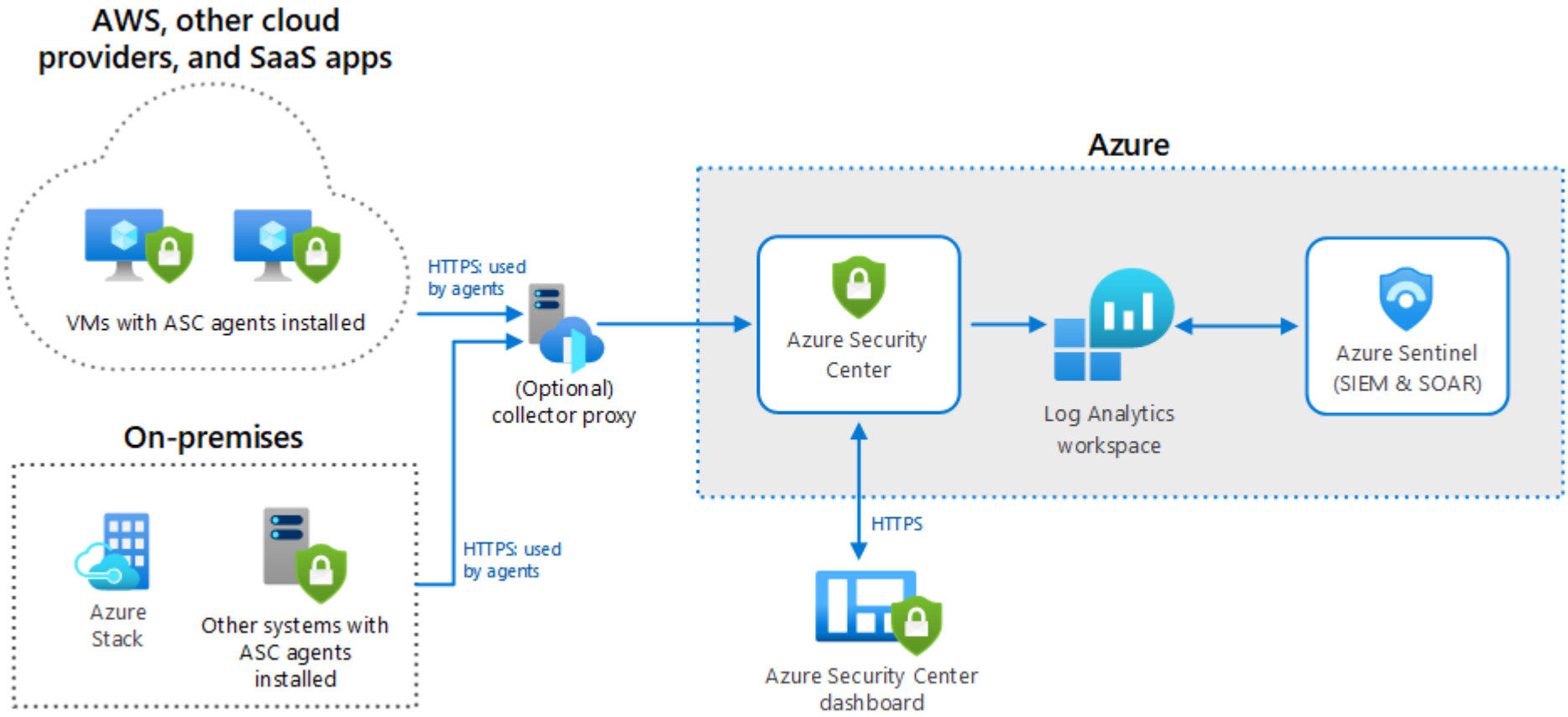


- Azure Key Vault (very good indeed)
  https://docs.microsoft.com/en-us/azure/key-vault/general/basic-concepts





- Azure Security Center & hybrid security monitoring (not covered further here)
  see https://docs.microsoft.com/en-us/azure/architecture/hybrid/hybrid-security-monitoring

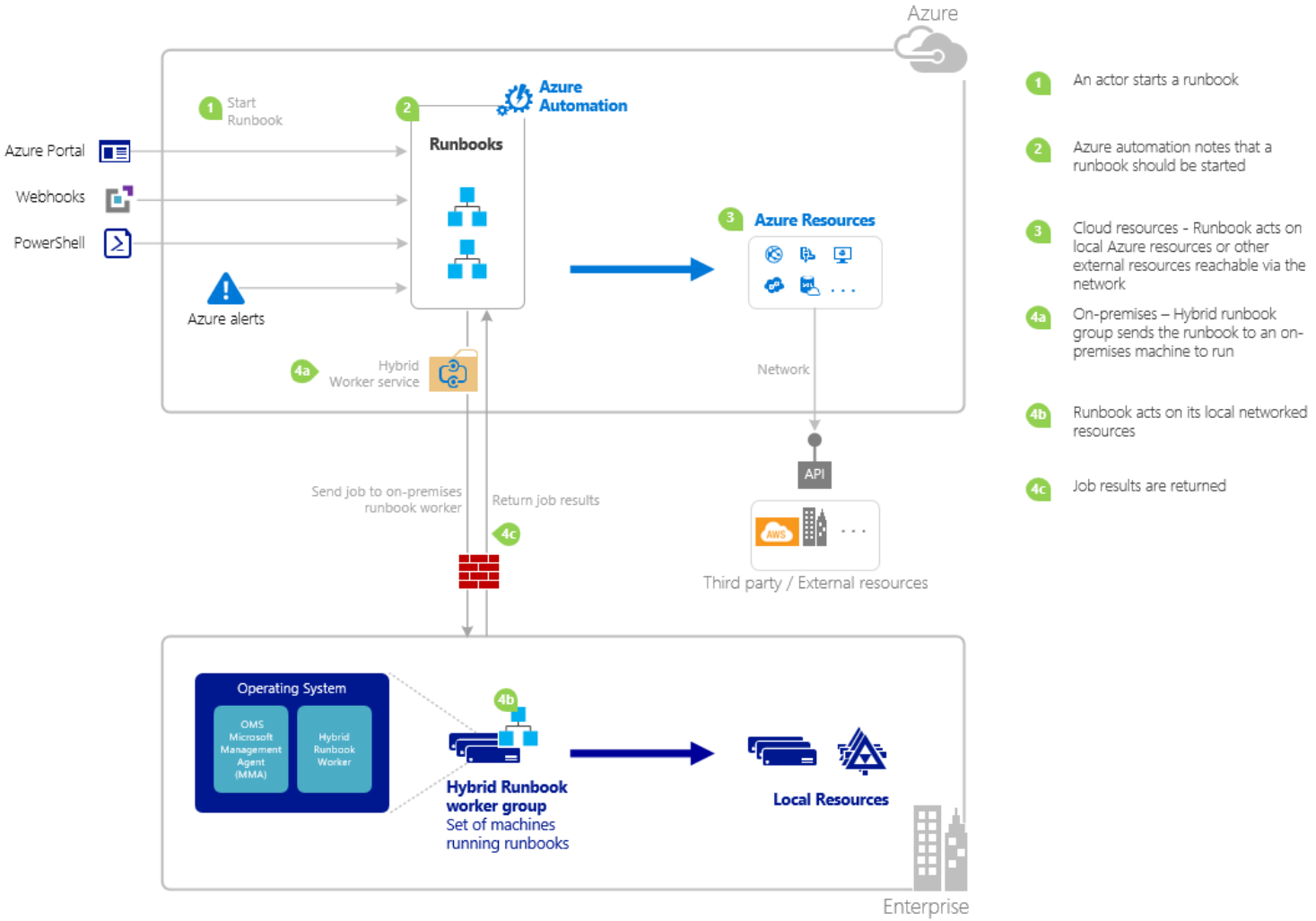


Excursion: The Bourne again shell (bash)

See https://tldp.org/LDP/Bash-Beginners-Guide/html/chap_01.html

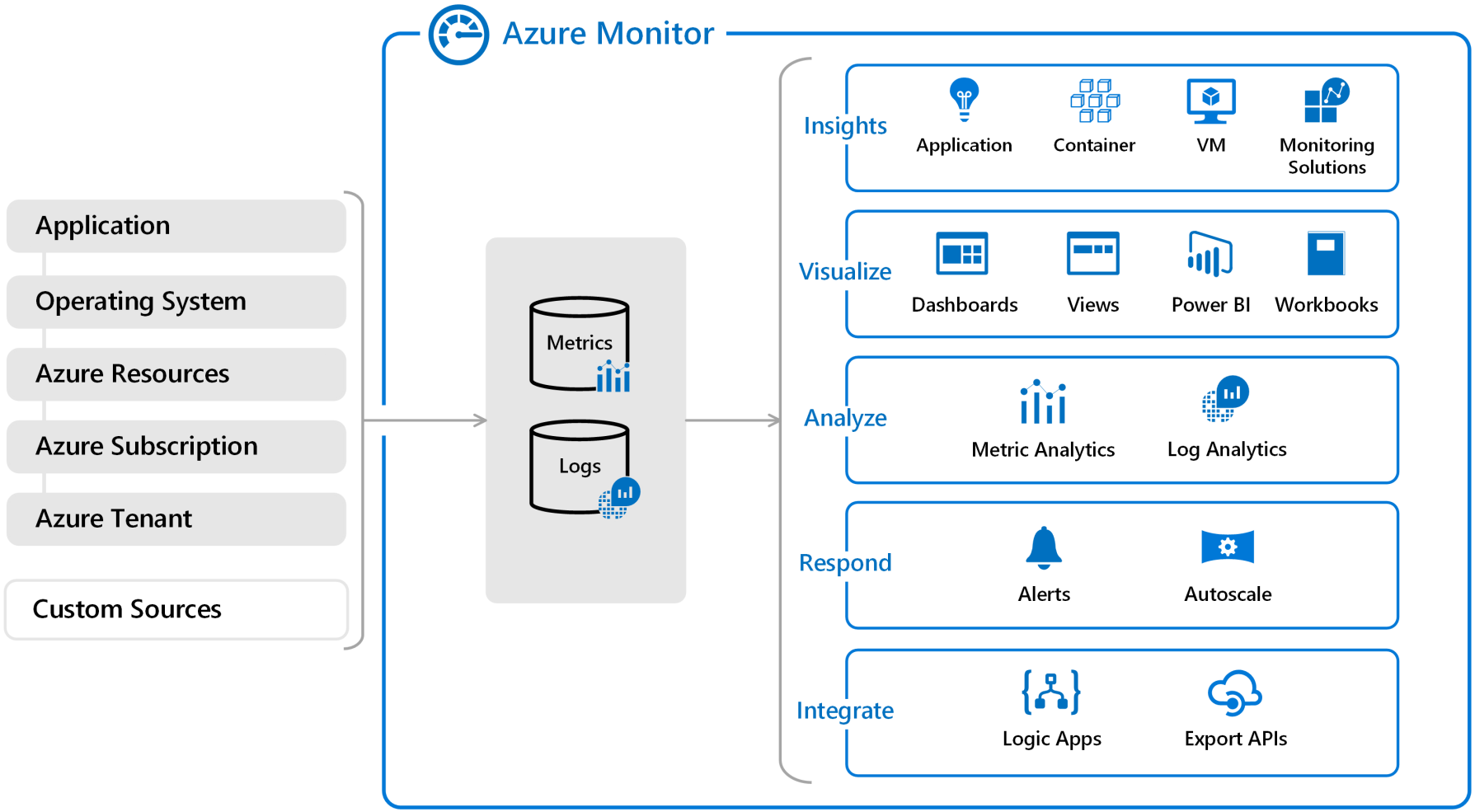
Azure Automation (not covered further here)

See https://docs.microsoft.com/en-us/azure/automation/automation-intro



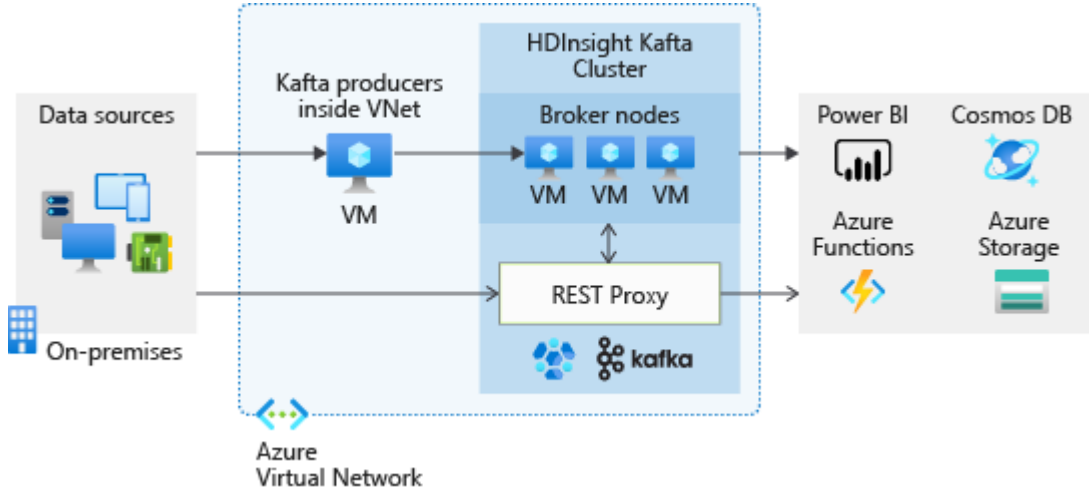
Azure Monitoring (not covered further here)

See https://docs.microsoft.com/en-us/azure/azure-monitor/overview



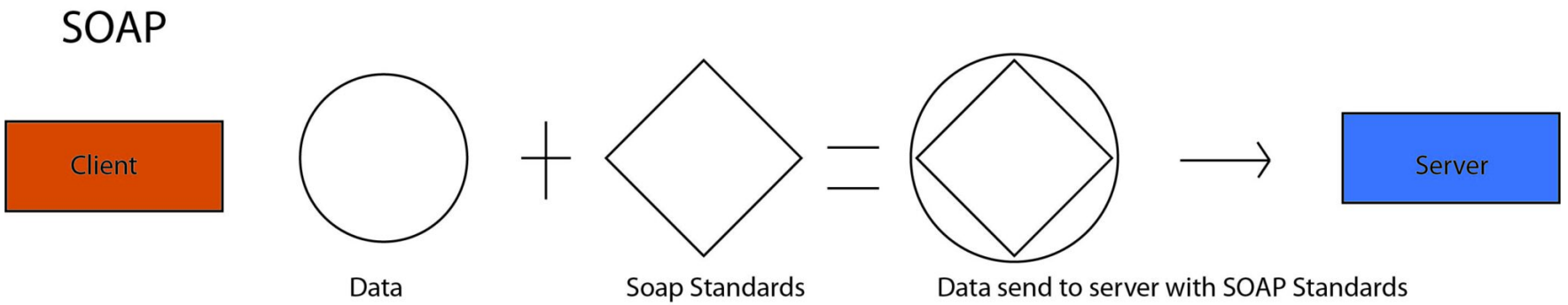
APIs 4 Azure

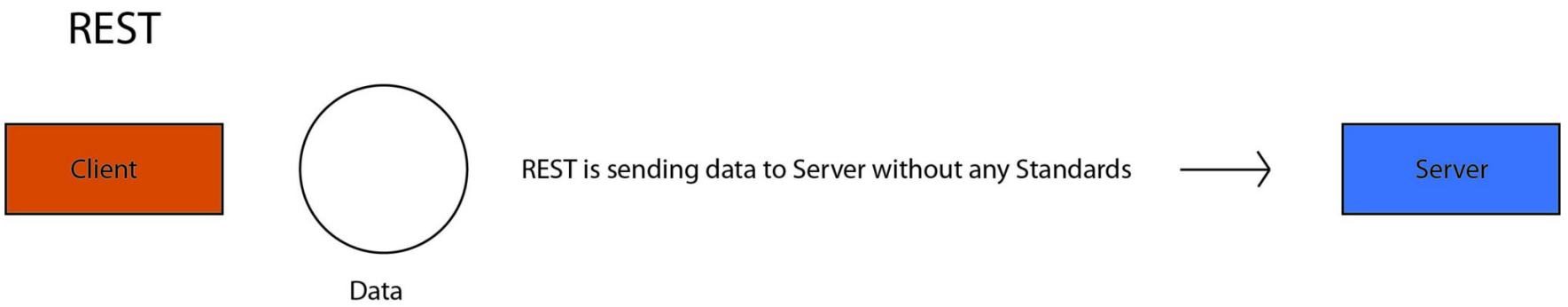
For communication between services, azure functions,VMs and Containers Azure heavily leans on REST-APIs. Though legacy SOAP-APIs are widely available, still. With these APIs you can also connect custom functions and applications (meaning your application, too) to arbitrary Azure services and functions. See https://docs.microsoft.com/en-us/rest/api/azure/



Relational State Transfer (REST) vs Simple Object Access Protocol (SOAP)

See https://en.wikipedia.org/wiki/Representational_state_transfer , https://en.wikipedia.org/wiki/SOAP

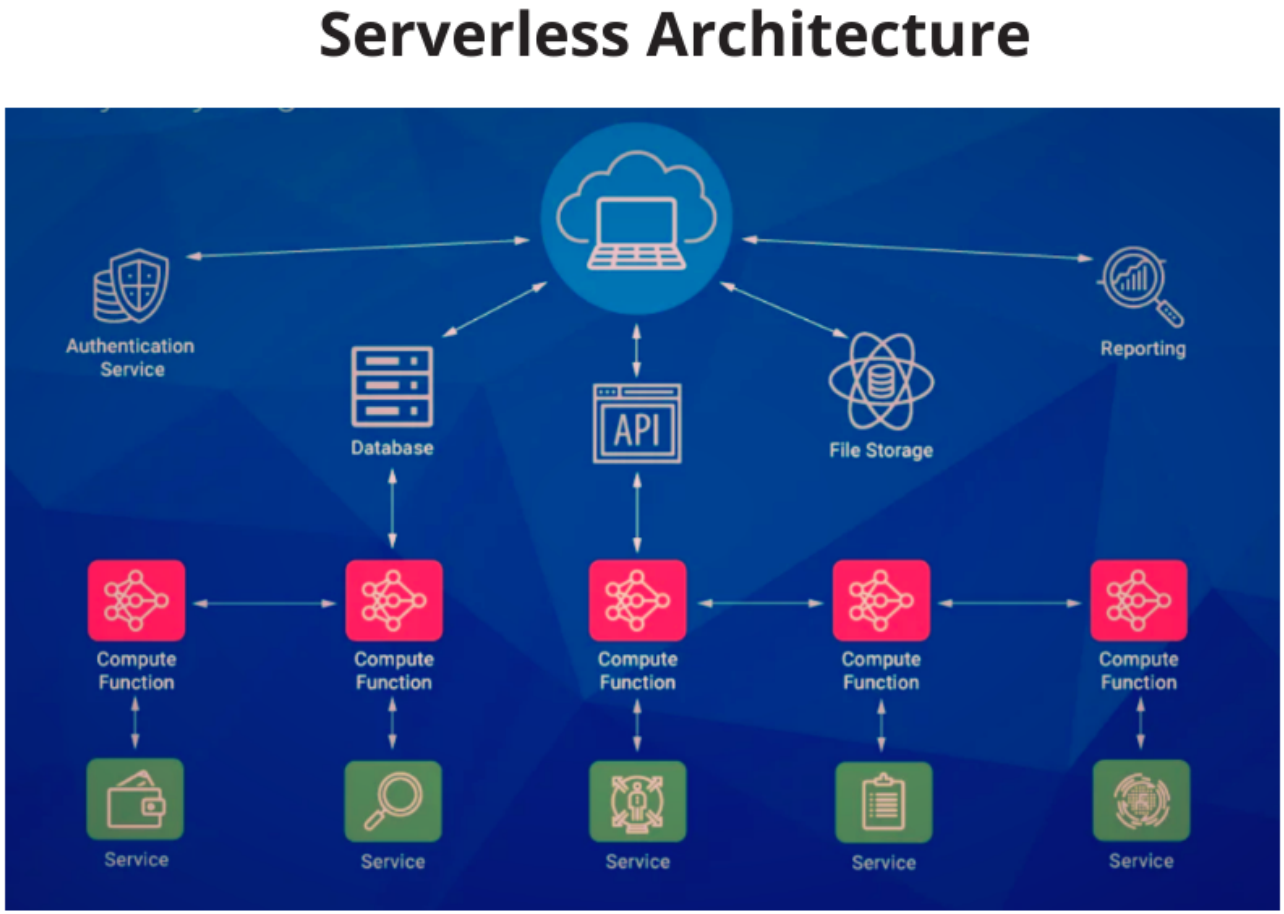




SOAP standards => whole object representation will be transmitted. That may be quite large.
REST has no standard => only netto data will be transmitted.

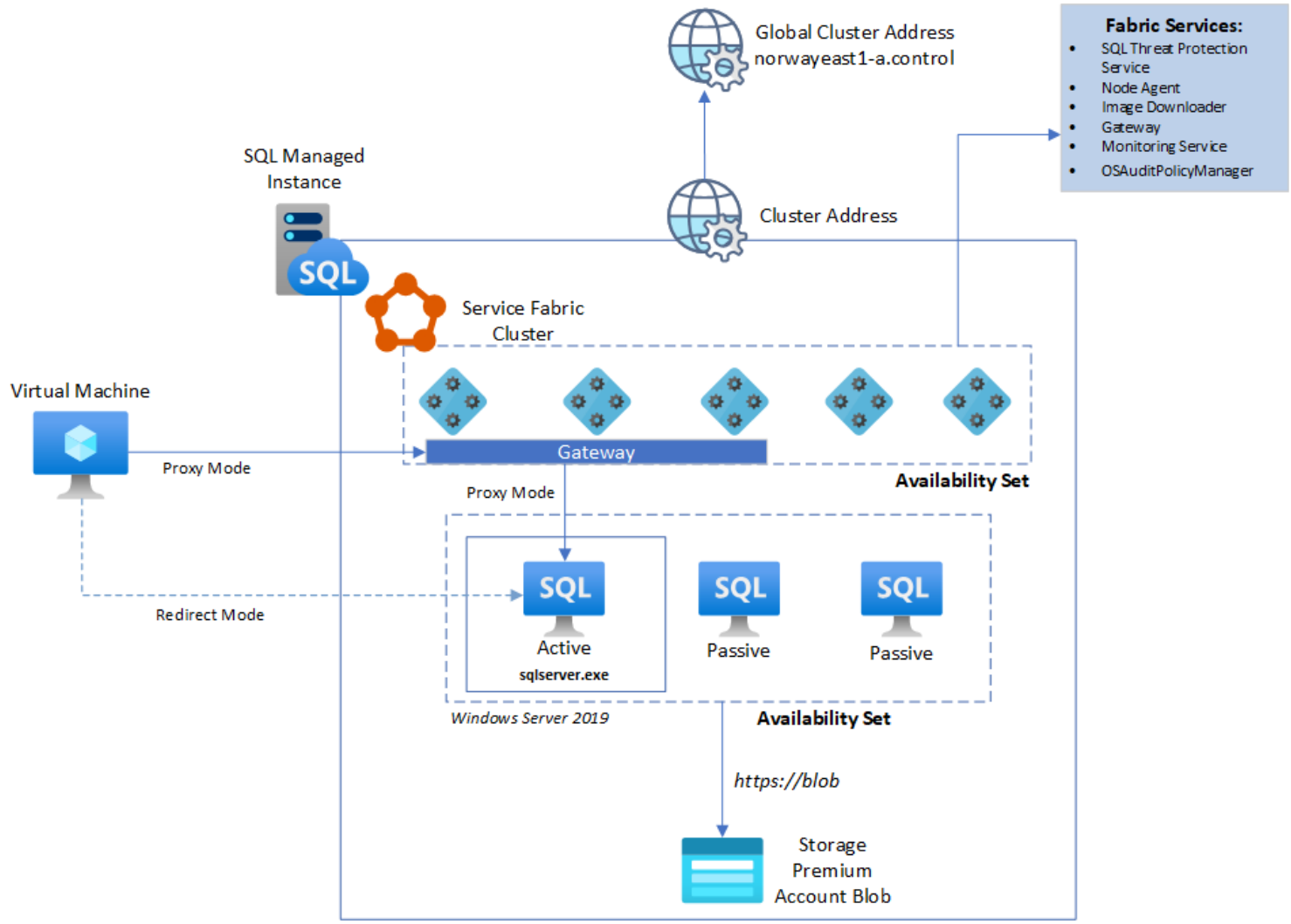
Both are stateless!

Azure Serverless Computing

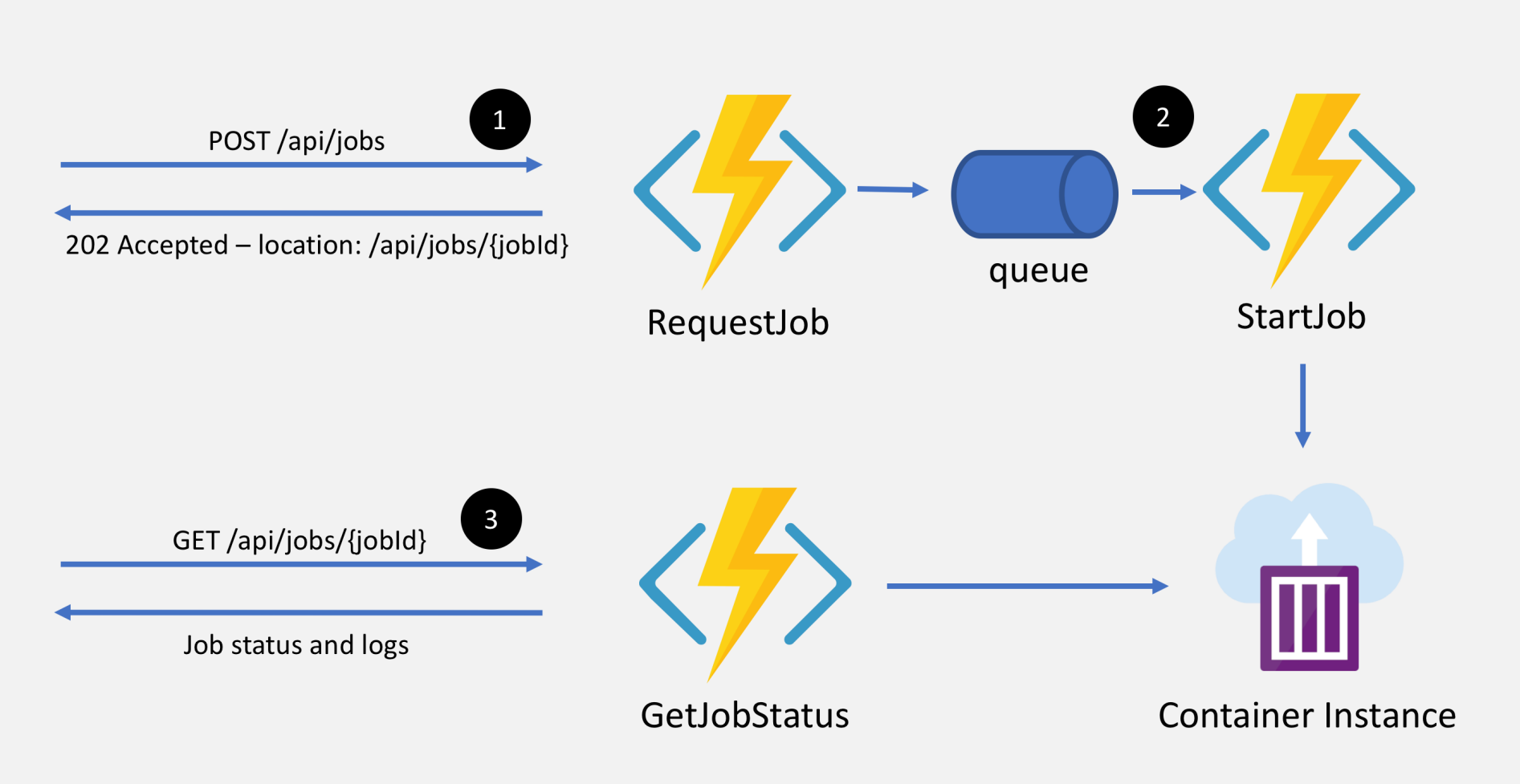


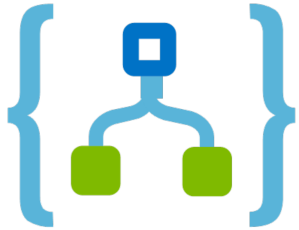
Features

- Fully managed instances. Example: Logical SQL-Server

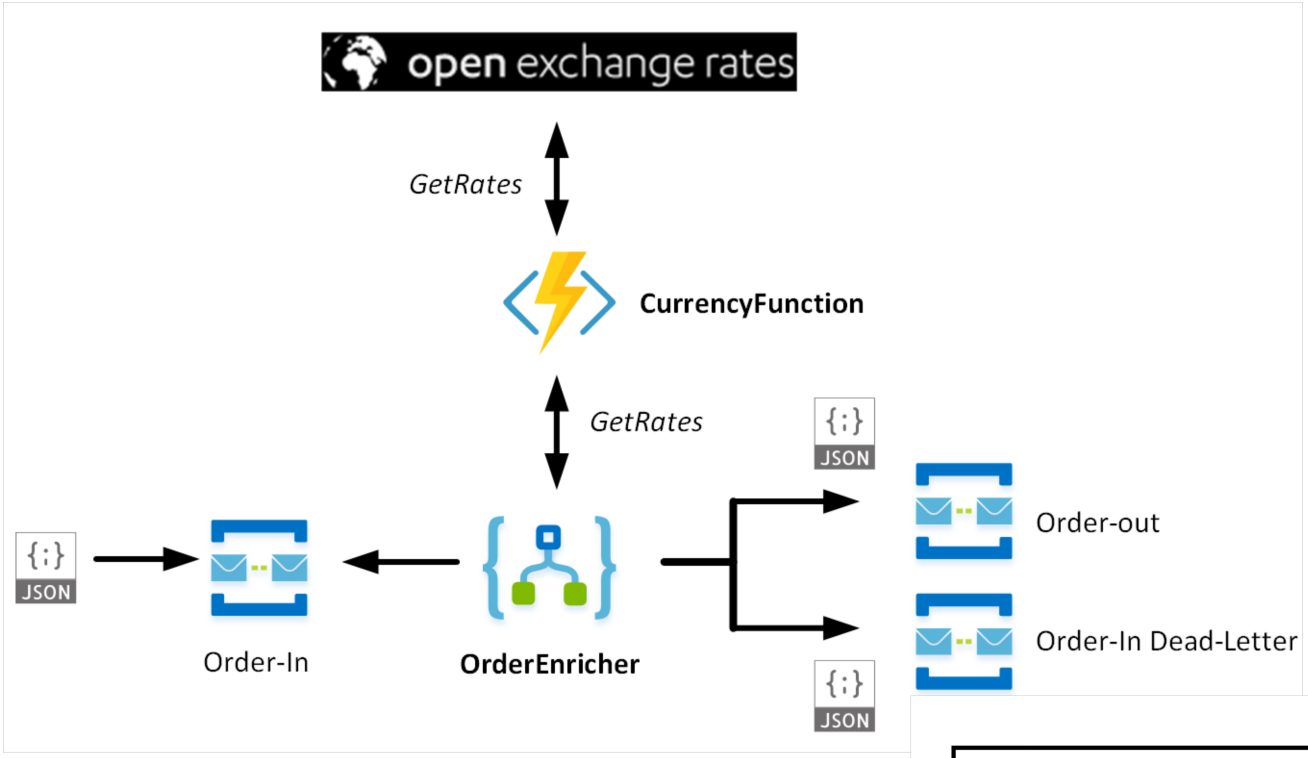


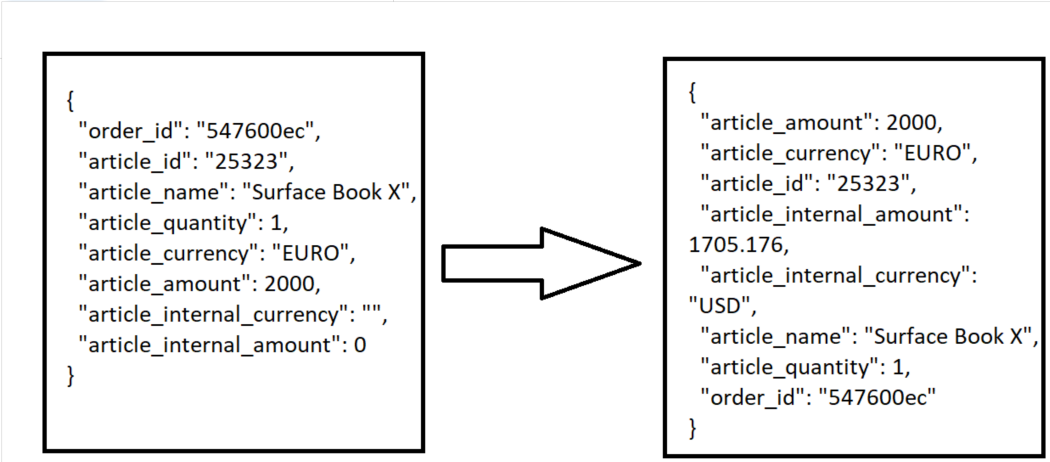
- Azure Functions

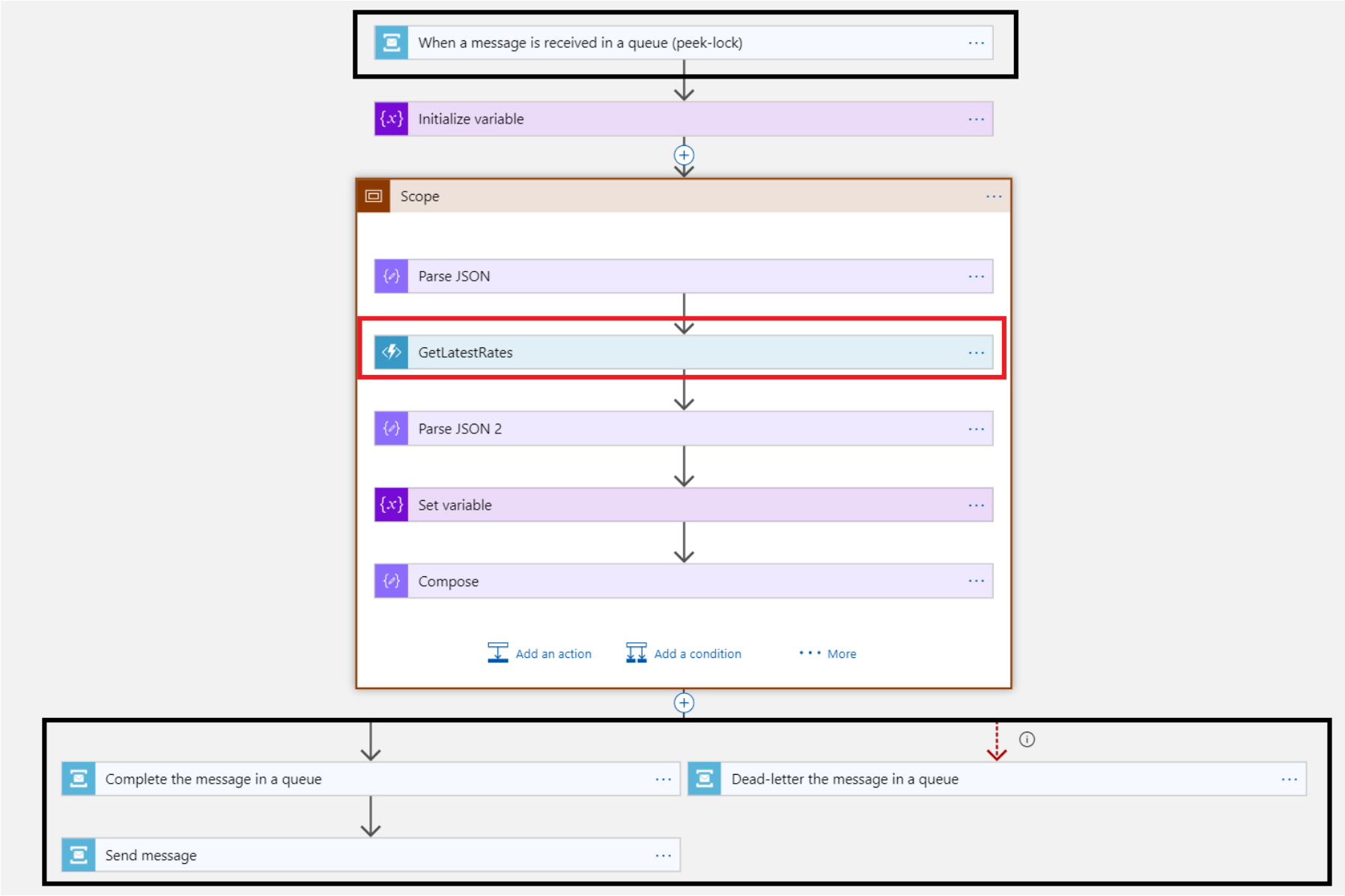




- Azure Logic Apps






When a message is received in a queue (peek-lock)
Initialize variable
Scope
Parse JSON
GetLatestRates
Parse JSON 2
Set variable
Compose
Add an action
Add a condition
More
Complete the message in a queue
Dead-letter the message in a queue
Send message

When to use Azure Logic Apps

Logic Apps are “event-driven” meaning they run based on a trigger. That is an HTTP request, message on a queue, a blob created in a container, or schedule. Typical scenarios for Logic Apps are:

- SaaS event processing
- Timer-based processing
- Data Ingestion
- Business Process
- Integration between (Cloud) services
- Content-based routing
- Data transformation and enrichment

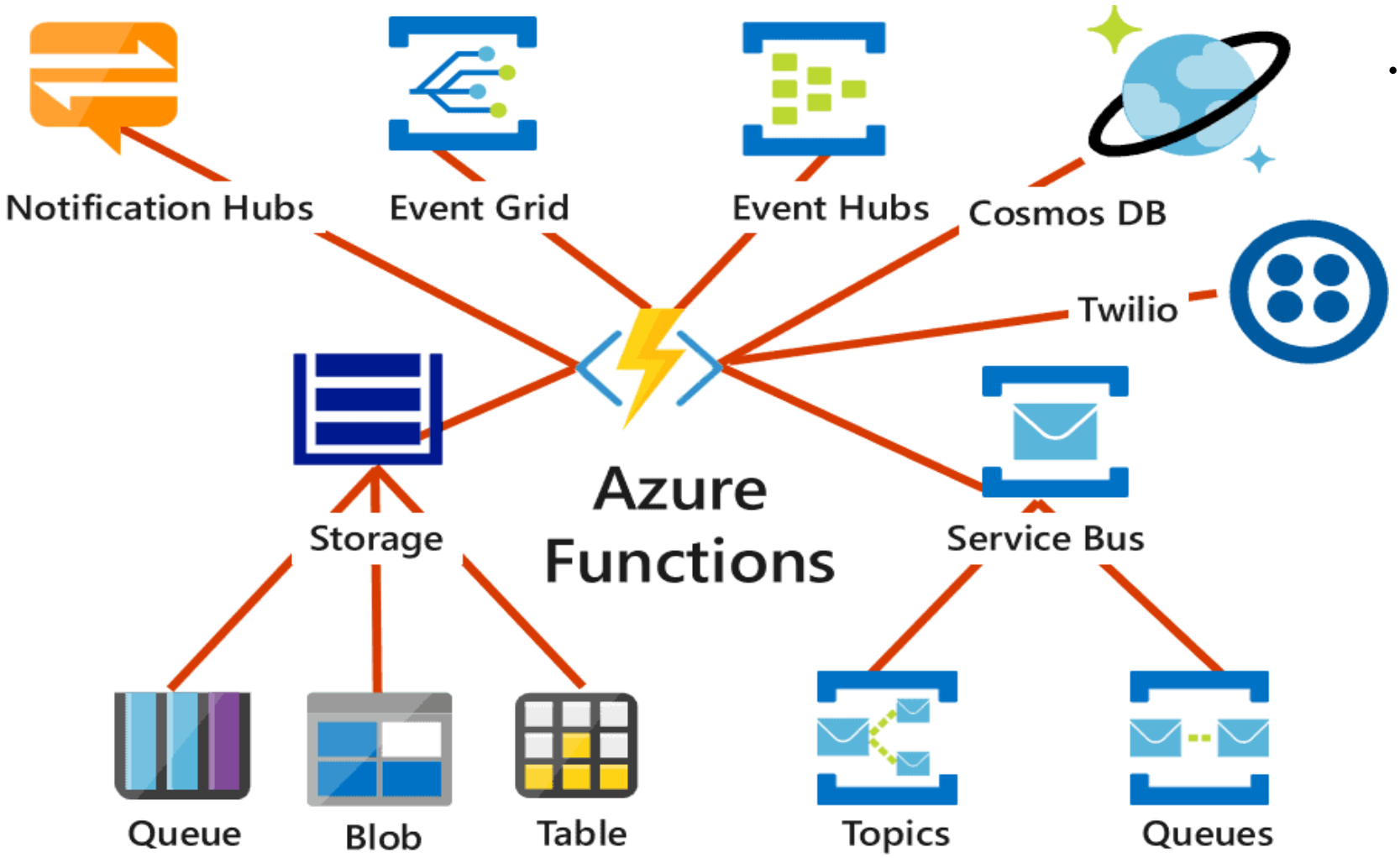
When to use Azure Functions

Azure Functions like Logic Apps are “event-driven” meaning they run based on associated and configured events, or “triggers”. Azure Functions can also respond to Azure-specific events. Such as an image added to a Storage Blob or a notification arriving in a Message Queue. Typical scenarios for Azure Functions are:

- Timer-based processing
- Azure service event processing
- SaaS event processing
- Serverless web application architectures
- Serverless mobile backends
- Real-time stream processing
- Real-time bot messaging

Azure Serverless Messaging and event dispatch

See https://docs.microsoft.com/en-us/azure/azure-functions/functions-triggers-bindings?tabs=csharp#overview



Choose between Azure messaging and event services - Event Grid, Event Hubs, and Service Bus

See https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services

Event vs. message services

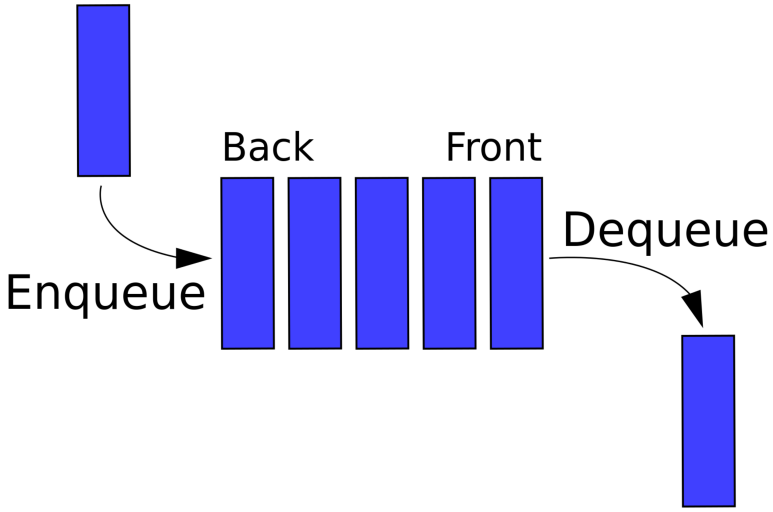
An event is a lightweight notification of a condition or a state change. The publisher of the event has no expectation about how the event is handled. The consumer of the event decides what to do with the notification.

A message is raw data produced by a service to be consumed or stored elsewhere. The message contains the data that triggered a message pipeline event. The publisher of the message has an expectation about how the consumer handles the message.

COMPARISON OF SERVICES

Service	Purpose	Type	When to use
Event Grid	Reactive programming	Event distribution (discrete)	React to status changes
Event Hubs	Big data pipeline	Event streaming (series)	Telemetry and distributed data streaming
Service Bus	High-value enterprise messaging	Message	Order processing and financial transactions

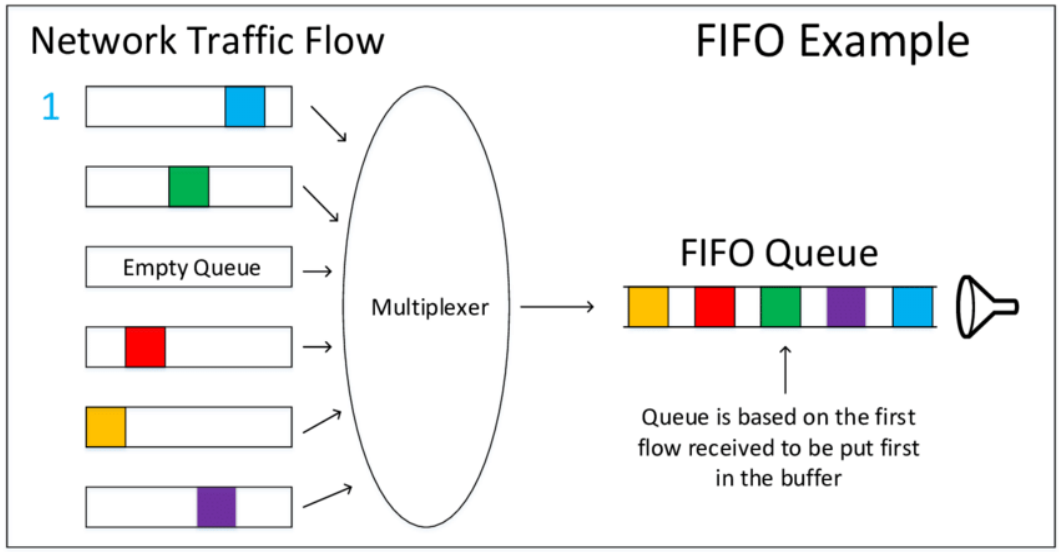
Service Bus – Queues



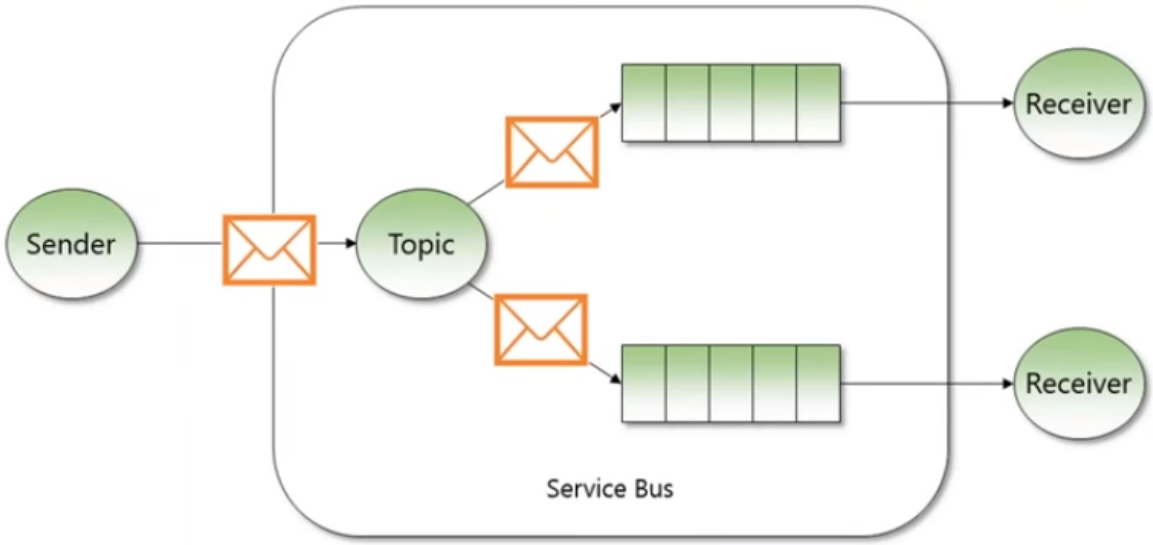
Queue Storage concepts

- Queues in Azure are FIFO
- Queue storage is persistent
- Messages are removed imediately after being red
- Unread messages are removed after TTL expires
- Peek ahead is always possible
- Queues can generate events when something arrives. To receive such an event one must register with the queue.
- Caveat: Events are only generated if the queue is empty and something arrives. The receiver is responsible for emptying queues before waiting for a new event.

- Many Instances can write to a queue with the help of a Mux
- In Azure you have to build the Mux all by yourself. F.e. with an Azure function
- The same is true for Azure Topics and multiple senders. See below.



Service Bus – Topics



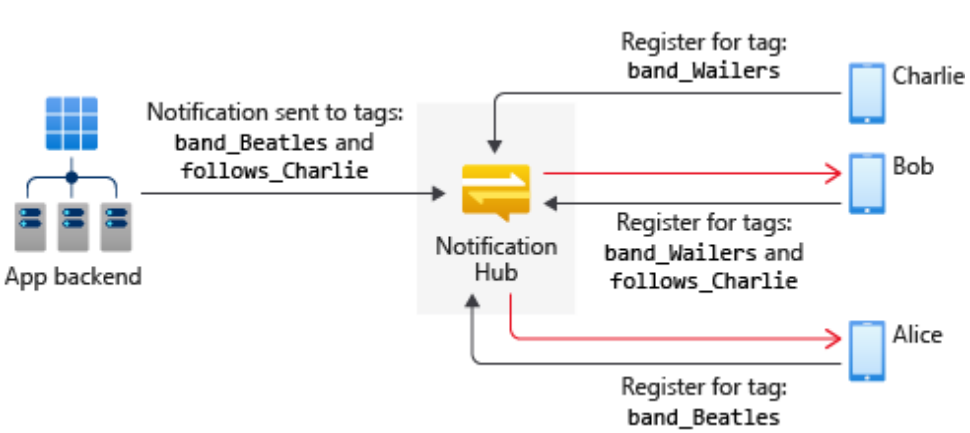
Topic concepts

- A Topic is an outgoing Mux
- A queue allows processing of a message by a single consumer. In contrast to queues, topics and subscriptions provide a one-to-many form of communication in a publish and subscribe pattern.
- A topic subscription resembles a virtual queue that receives copies of the messages that are sent to the topic. Consumers receive messages from a subscription identically to the way they receive messages from a queue.

Please see discussion of *Service Bus – Queues* above for a discussion of multiple senders.

Notification hubs

See https://docs.microsoft.com/en-us/azure/notification-hubs/notification-hubs-tags-segment-push-message



Notification Hub concepts

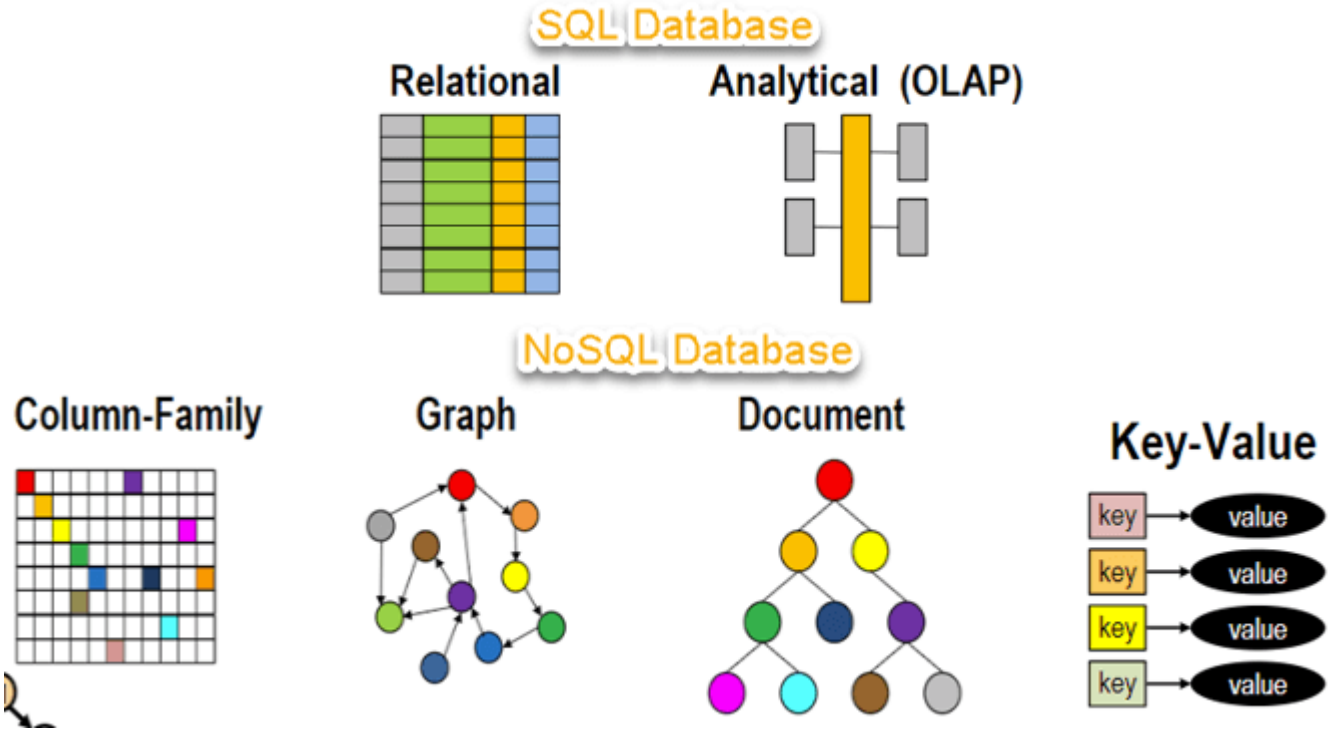
- In principle Notification Hubs work like Topics (see above)
- Contrary to Topics messages can be filtered by using tags

The only way to target specific notification registrations is to associate tags with them, then target those tags. To receive push notifications, an app must register a device handle on a notification hub. Once the app creates a registration on a notification hub, the application backend can send push notifications to it. The application backend can choose the registrations to target with a specific notification in the following ways:

1. **Broadcast**: all registrations in the notification hub receive the notification.
2. **Tag**: all registrations that contain the specified tag receive the notification.
3. **Tag expression**: all registrations whose set of tags match the specified expression receive the notification.

Cosmos DB

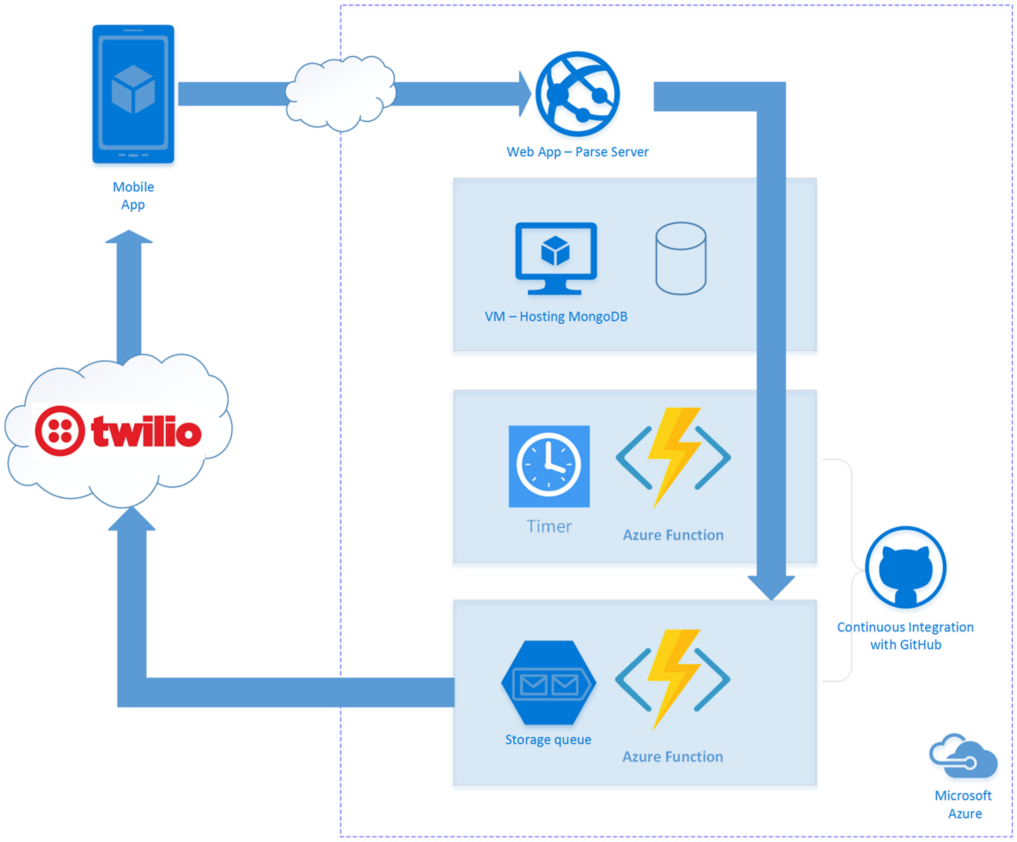
Cosmos DB is an interface to four different types of NoSQL (better: NotRelational) databases. Arbitrary events can produce events for which an Azure function or Azure Logic App can register.



See https://www.guru99.com/nosql-tutorial.html a very good tutorial on the topic of SQL vs. NoSQL databases

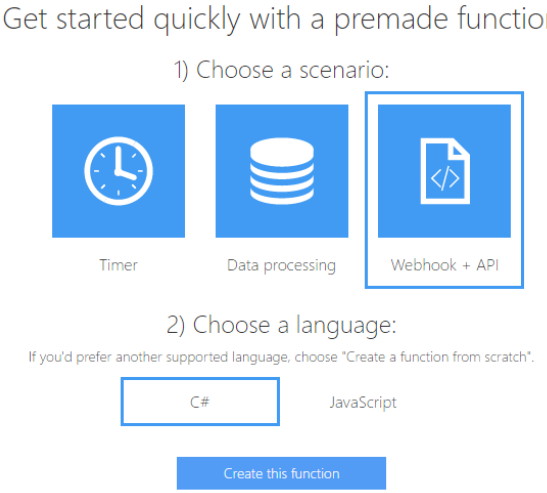
Twilio: A proprietary cloud communications platform as service (PaaS)

See https://en.wikipedia.org/wiki/Twilio



Twilio allows software developers to programmatically make and receive phone calls, send and receive text messages, and perform other communication functions using its web service APIs.

It offers Serverless Webhooks with Azure Functions and multiple programming languages.

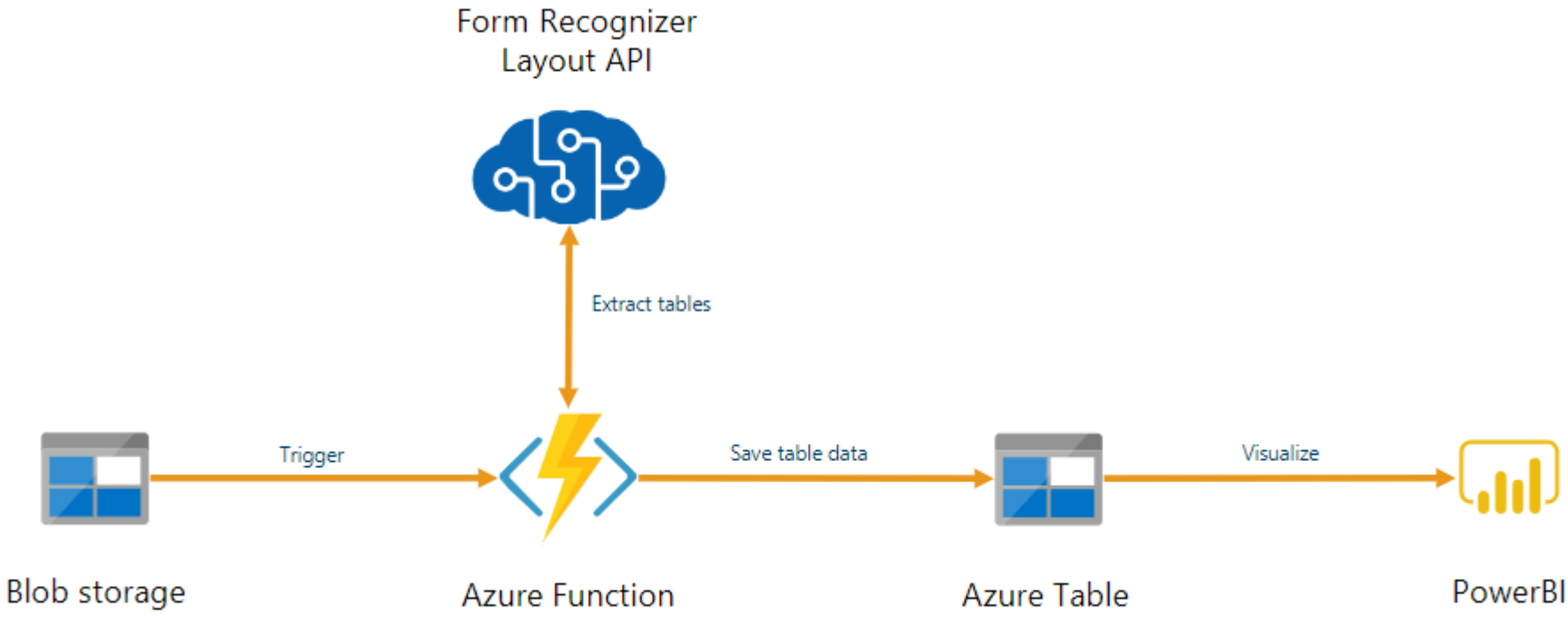


Messages and events from storage

Like queue storage Blob and table storage can produce events, too.

See https://docs.microsoft.com/en-us/azure/azure-functions/functions-triggers-bindings?tabs=csharp#overview

See https://docs.microsoft.com/en-us/azure/applied-ai-services/form-recognizer/tutorial-azure-function



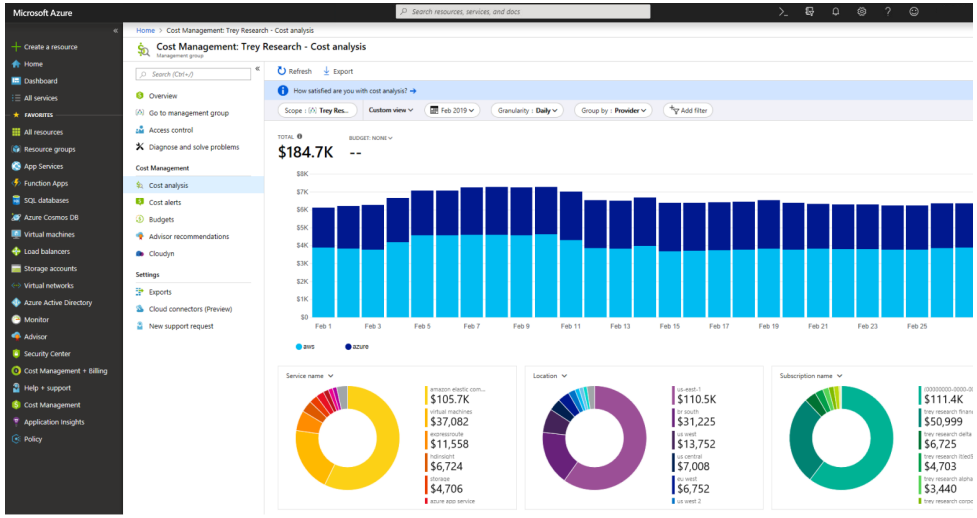
Azure Pricing

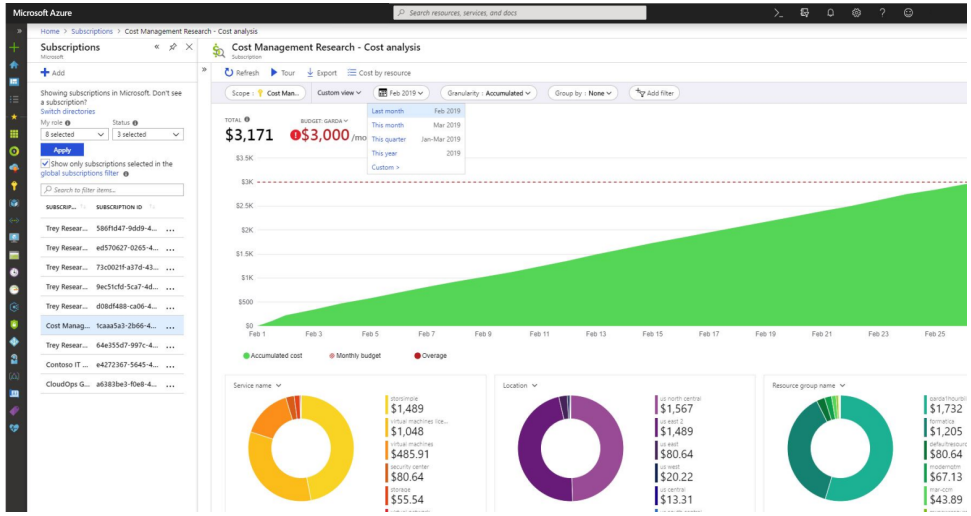
Azure pricing is quite complicated. Even more so because services may or may not be available depending on location, license, storage tier, and performance tier. Furthermore, remember that generally spoken only continental European locations conform to European data protection laws. There is an Azure Pricing Calculator (https://azure.microsoft.com/en-us/pricing/calculator/) and the Azure Pricing Overview (https://azure.microsoft.com/en-us/pricing/). But both are unreliable, error prone, and often out of date.

The only way is to experiment and closely monitor amortized costs versus expected costs.

Azure Cost Monitoring

There are the usual GUI-graphics. See https://azure.microsoft.com/de-de/services/cost-management/





But what you really should do is Automated Cost Reporting (with alerts).

Automated Cost Reporting (with alerts)

See https://docs.microsoft.com/en-us/azure/cost-management-billing/manage/cost-management-automation-scenarios

See https://docs.microsoft.com/en-us/azure/cost-management-billing/costs/tutorial-export-acm-data?tabs=azure-portal

See https://mertsenel.tech/project/azure-usage-and-cost-tracking-durable-functions-powershell/