Application Modernization

AUGUST 27, 2018 | 6-9 PM | NOKIA: 6000 CONNECTION DRIVE, IRVING, TX, 75039

DAVE WEILERT , EVAN HAMAMCY, PAT FRUTH

Agenda

- What is Application Modernization
- The many “Re’s”
- People, Product, Process
- Microservices
- Containers
- Container Orchestration
- Serverless
- CI/CD
- What’s next
- Service mesh
- Questions / Comments
- Hands on labs

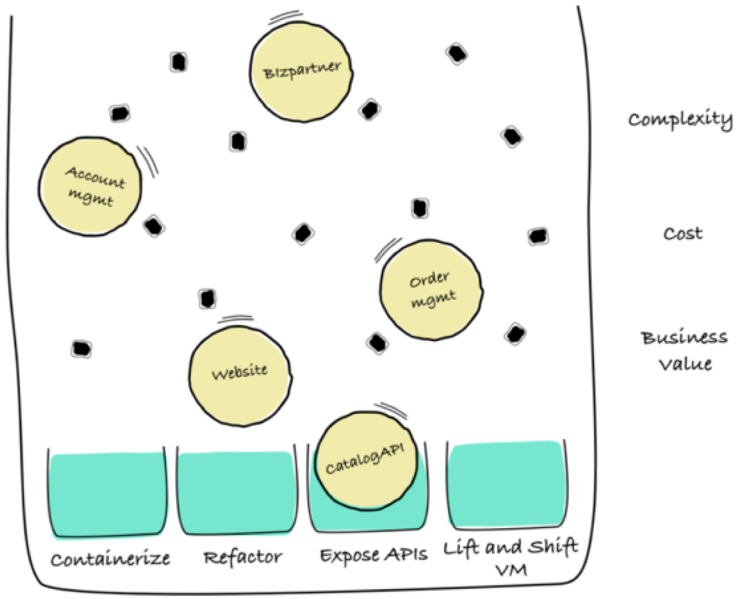
Application Modernization

. . . ***align it more closely with current business needs***.

The overall goal of a modernization project is to ***"create new value from existing applications."***

https://www.glowtouch.com/what-is-application-modernization/

Application modernization “depends on”



- What
- Who
- When
- Why
- Where

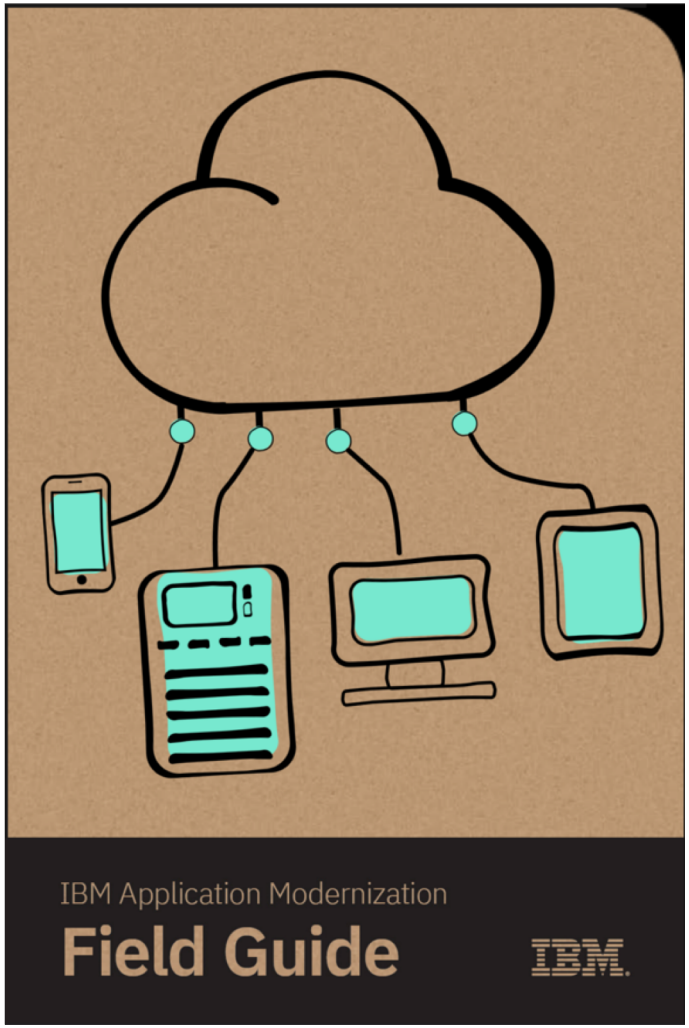
Very much like a great story

https://www.ibm.com/cloud/garage/content/culture/app-modernization-field-guide/

Application Modernization

IBM

approach

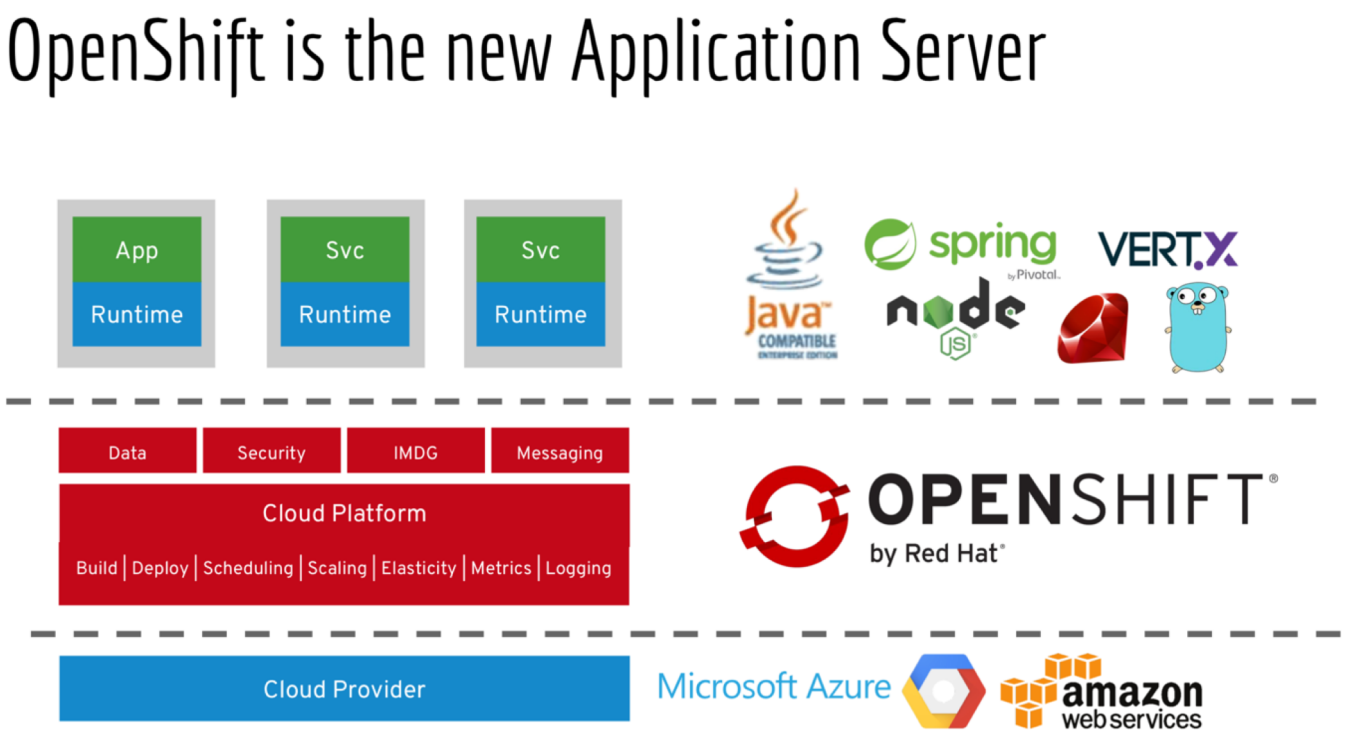


https://www.ibm.com/cloud/garage/content/culture/app-modernization-field-guide/

Application Modernization

Red Hat

approach



Application Modernization

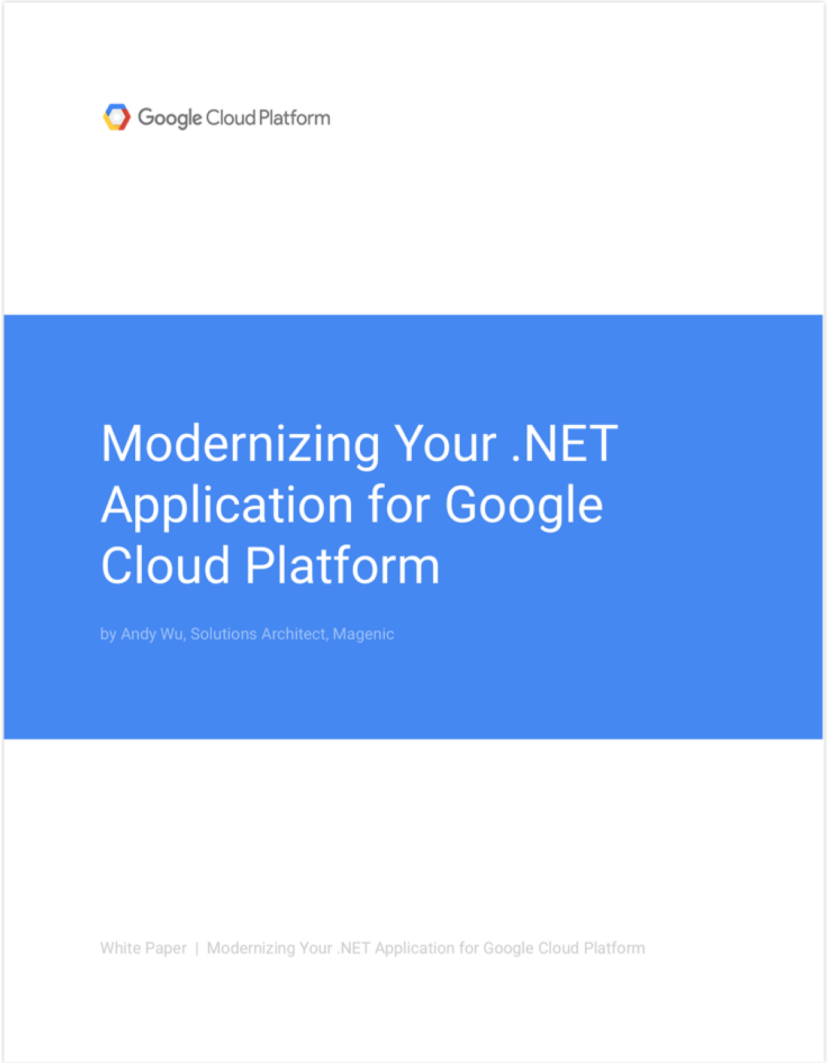
Dell, VMware, Pivotal

approach

Application Modernization

Google with .NET

approach



Application Modernization

Cognizant

approach

COGNIZANT

Cognizant 20-20 Insights

Legacy Enterprise Systems Modernization: Five Ways of Responding to Market Forces

Whether your organization replaces the entire enterprise application system, replaces one application/component at a time, applies localized fixes in one application or modernizes underlying infrastructure, any approach is typically better than none at all.

Executive Summary

Market forces demand a response from companies in every corner of the world seeking to maintain a modern and effective enterprise information environment. Whether it is consumerism, digitization, social-mobile-analytics-cloud (aka the SMAC Stack) or productization, the organizations that run business IT systems must be agile enough to respond to market demands in better-faster-cheaper ways.

However, not all legacy enterprise applications - many of which were built on decades-old technology and use architectures that were built when IBM mainframes still ruled - can support the continuously accelerating pace of business change.

Hence, many companies now realize that they must transform their IT systems with a modern technology stack or risk irrelevance.

However, how they approach this transformation depends on factors such as their organizational culture, scope, urgency, risk appetite, existing technology stack, technical expertise, program management maturity, funding availability, etc.

This white paper describes the five ways we have seen companies approach legacy modernization, four of which, if done right, can keep them in the competition.

cognizant 20-20 insights | july 2015

Cognizant

Application Modernization

By 2022 US market could be over 16 billion annually

It’s big and it’s not going away!

The many ‘Re’s of Application Modernization

- Re-Platform
- Re-Host
- Re-Code
- Re-Write
- Re-Factor
- Re-Architect
- Re-Build
- RePlace
- ReTire
- ReTain

Multiple approaches to Application Modernization

People, Product, Process

People: Identify the relevant stakeholders of the SOURCE and TARGET applications. Identify the 'right' teams to do the job!!

Product: Identify the state the TARGET application needs to be in, making sure there are no functional gaps compared to SOURCE application

Process: The process to build the TARGET application is extremely important. Modernization projects are almost always going to change the 'running' production applications, and you are sure to hear a lot of voices saying 'DO NOT' try to change this or that! There is a fear factor involved that the process needs to overcome! So, lay out the process to the stakeholders (People,) before you do anything!

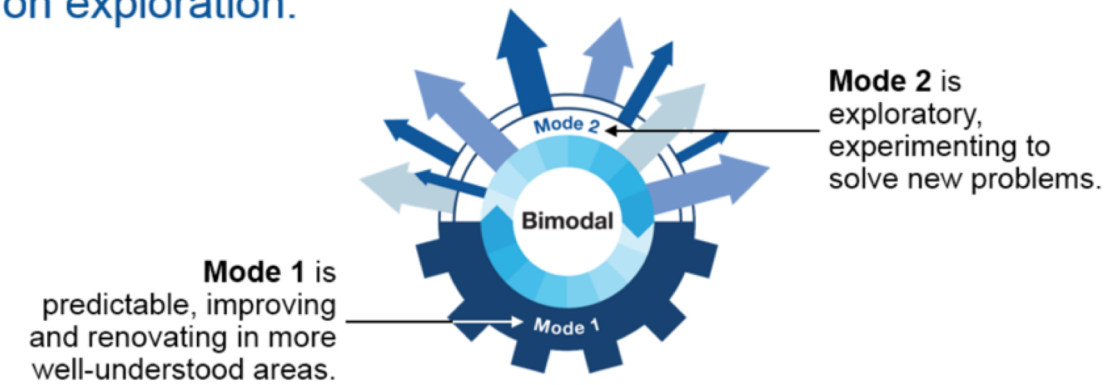
https://dzone.com/articles/application-modernizationmigration-what-why-when-and-how

Bimodal IT

Gartner IT Glossary > Bimodal

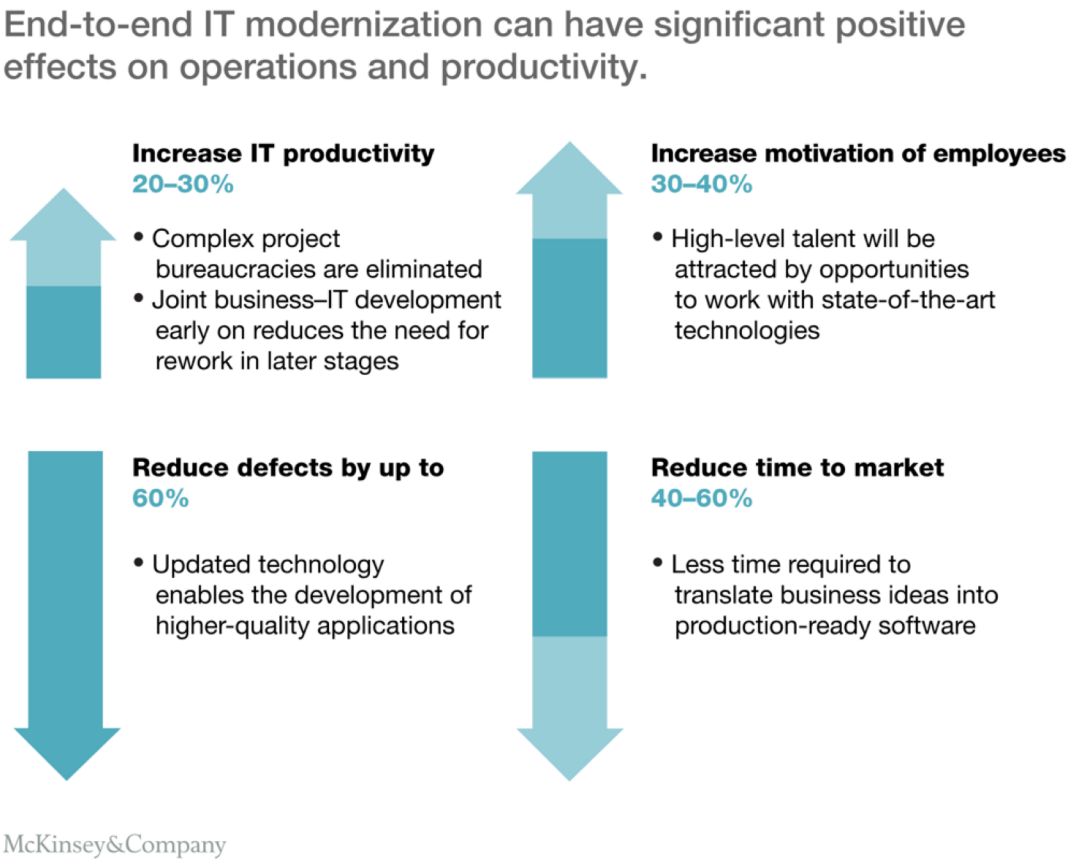
Bimodal

Bimodal is the practice of managing two separate but coherent styles of work — one focused on predictability and the other on exploration.



https://www.gartner.com/it-glossary/bimodal/

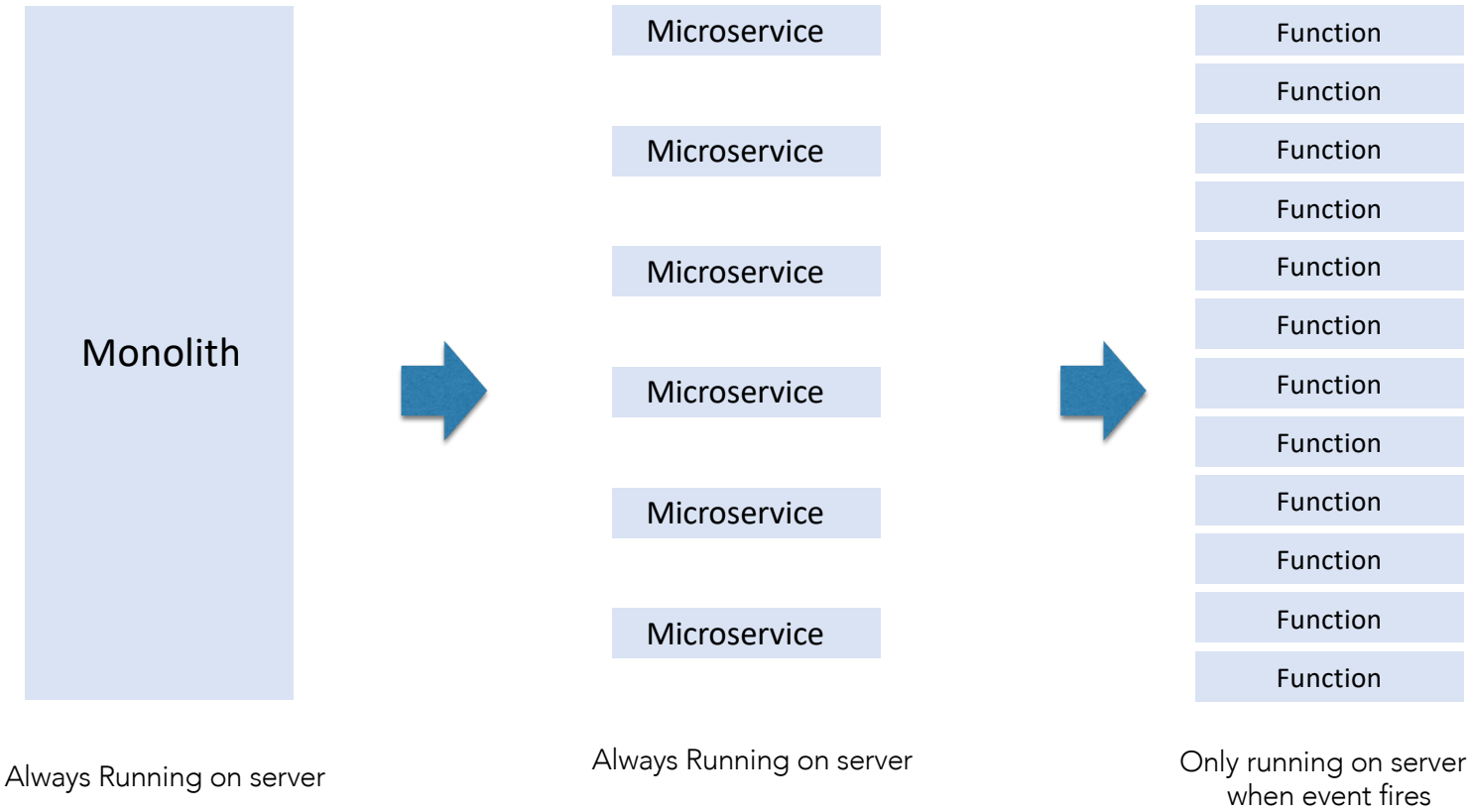
End-to-end IT modernization



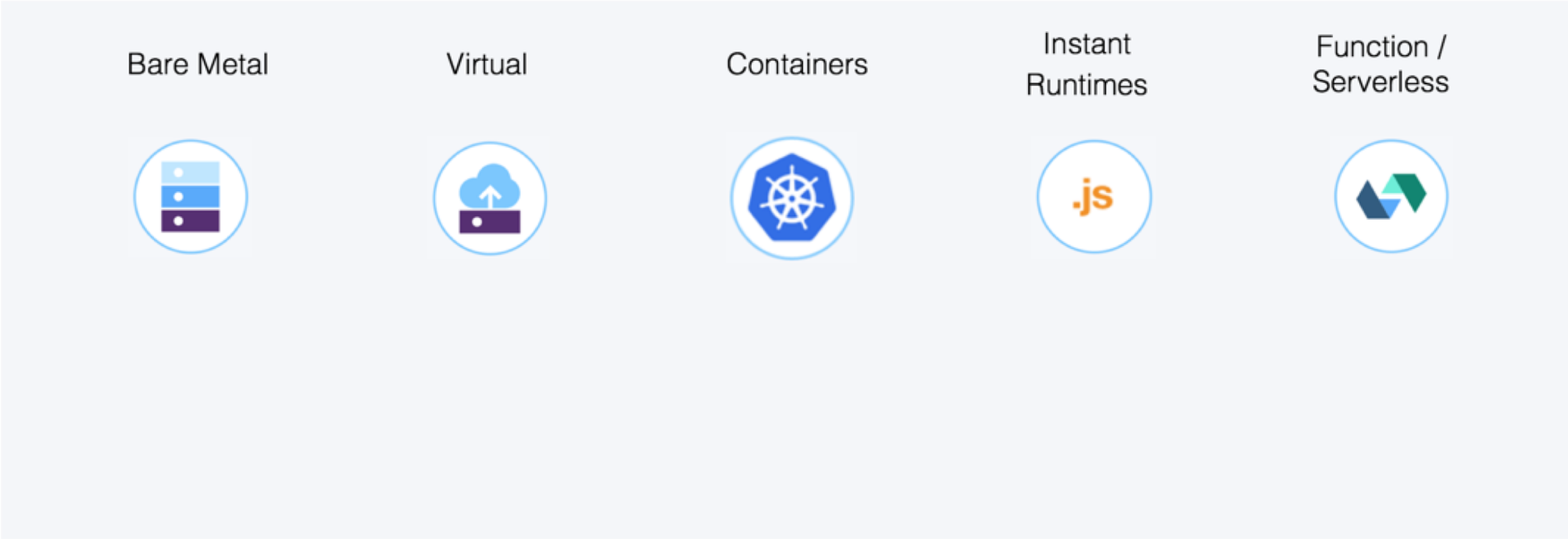
Compared with incrementalism, the end-to-end path toward a modern IT landscape can be more **risky**, and potentially more **expensive**.

https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/modernizing-it-for-a-digital-era

Application development strategies evolve



Application deployment strategies evolve



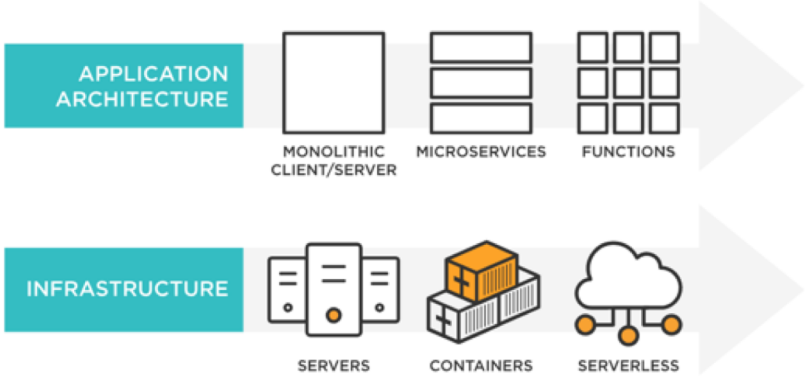
Customization; higher costs; slower time to value

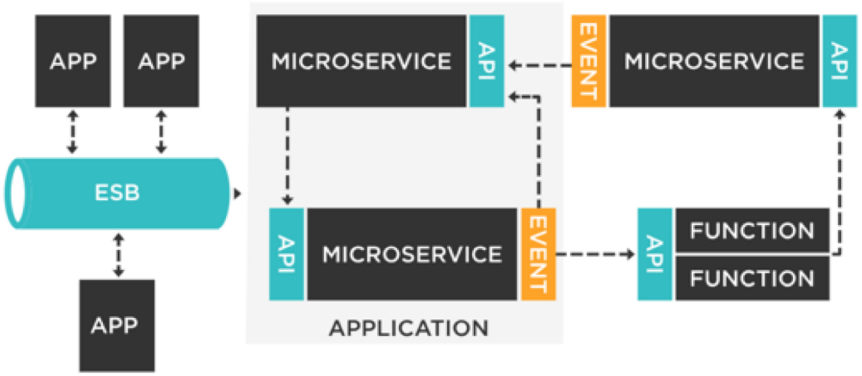
Control

Speed

Standardization; lower costs; faster time to value

Microservices and Serverless functions





https://www.tibco.com/solutions/microservices?utm_medium=cpc&utm_source=google&utm_term=%2Bmicro%20%2Bservices&utm_campaign=ggl_s_us_en_tci_microservices_beta&_bt=263683913936&_bm=b&_bn=g&gclid=CjwKCAjwwo7cBRBwEiwAMEoXPDdMhNXpvsj5d13jlLThvKB3hgqClaLsv9c3E

Six Mistakes to Avoid on Your Journey

#1: Failure to secure proper buy-in.

#2: Lack of understanding about your current state.

#3: Underestimating the importance of testing and planning.

#4: Ignoring the need to commit resources.

#5: Failing to maintain regular communications.

#6: Going it alone.

https://www.cio.com/article/2462476/enterprise-software/six-mistakes-to-avoid-on-your-modernization-journey.html

Microservices

Microservices Architecture

The term ***"Microservice Architecture"*** has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services. While there is no precise definition of this architectural style, there are certain common characteristics around organization around business capability, automated deployment, intelligence in the endpoints, and decentralized control of languages and data.

https://martinfowler.com/articles/microservices.html?cm_mc_uid=34013674245414737751113&cm_mc_sid_50200000=147499071

Microservices

One of the emerging trends is to engineer applications using ***microservices***.

Such an approach promotes developing and deploying applications that are composed of independent, autonomous, modular, self-contained units.

https://www.computerweekly.com/essentialguide/Essential-guide-to-application-modernisation

Benefits to Microservices

Applications built as a set of modular components are *easier* to ***understand***, easier to ***test***, and most importantly easier to ***maintain*** over the life of the application.

https://opensource.com/resources/what-are-microservices

Benefits to Microservices

One of the core benefits of developing new systems with microservices is that the architecture allows developers to build and modify individual components ***independently***

Five characteristics of a well-designed service

#1: It doesn't share database tables with another service

#2: It has a minimal amount of database tables

#3: It's thoughtfully stateful or stateless

#4: Its data availability needs are accounted for

#5: It's a single source of truth

https://opensource.com/article/18/4/guide-design-microservices

What's the hardest part about microservices?

Your data

https://opensource.com/article/17/5/hardest-part-about-microservices-your-data

“Goldilocks Principle”

Not too ***big***

Not too ***small***

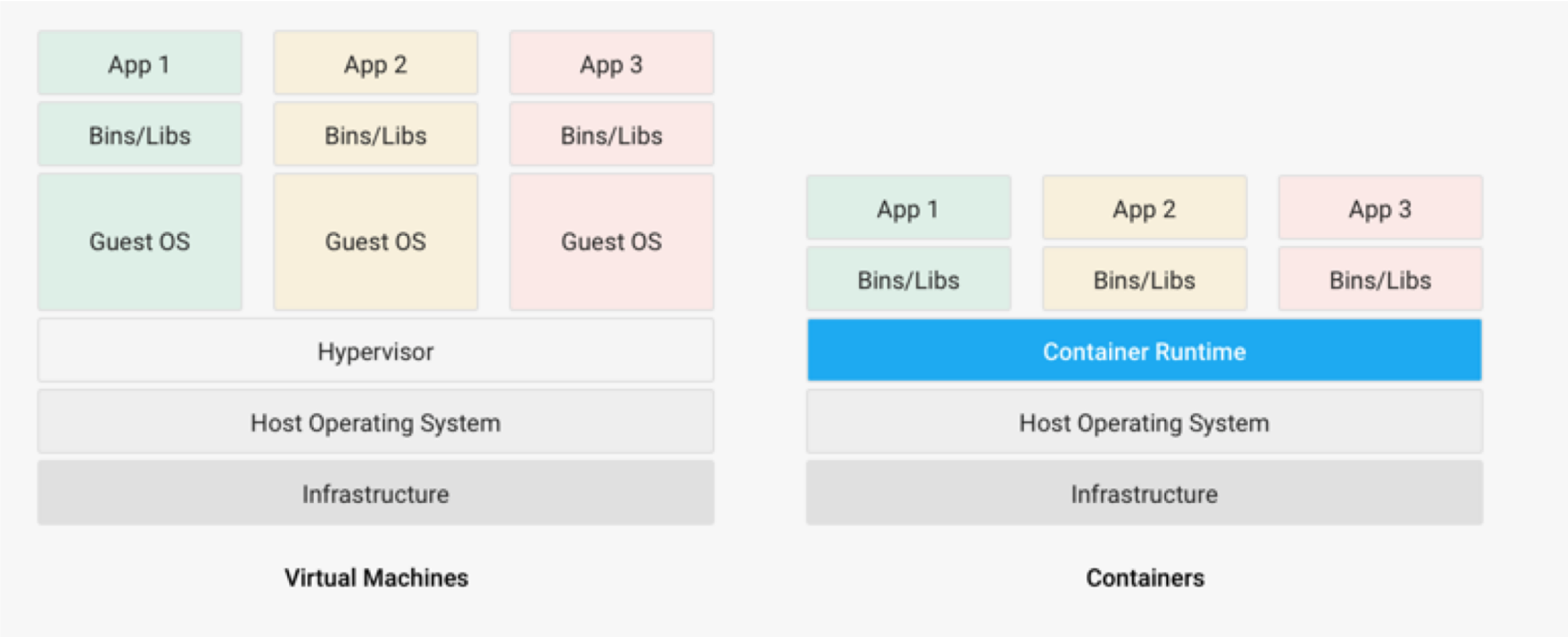
and

Not too ***tightly coupled***

https://opensource.com/article/18/4/guide-design-microservices

Containers

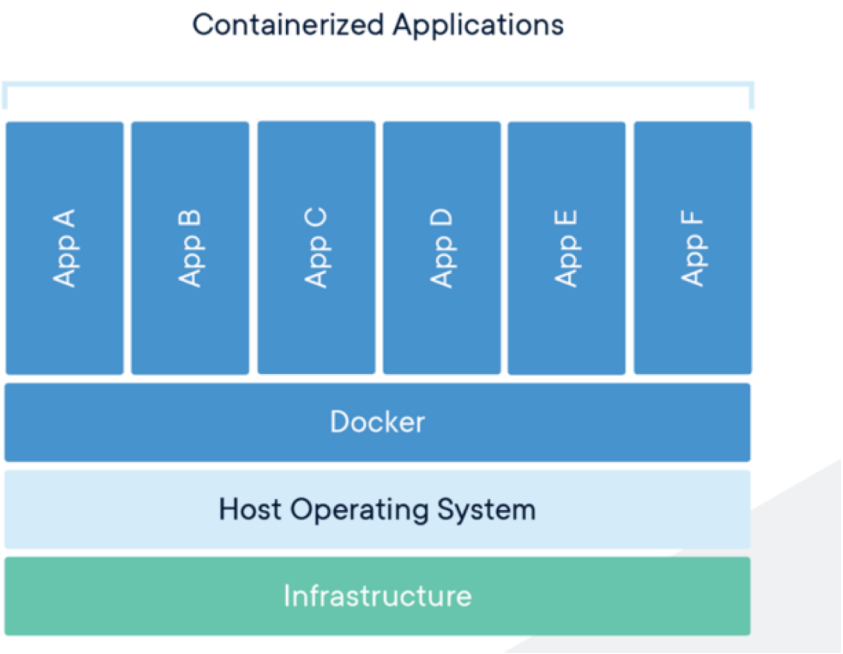
Containers according to Google



Containers offer a logical packaging mechanism in which applications can be abstracted from the environment in which they actually run.

https://cloud.google.com/containers/

Containers according to Docker



A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

https://www.docker.com/resources/what-container

Container Orchestration

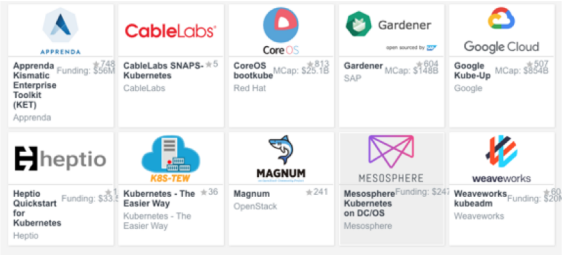
Container Orchestration

… there was a war waging for the standard in the Container Orchestration layer. Players like Apache Mesos Marathon, Docker Swarm, ***Kubernetes***, Openstack Magnum and VMware Photon were all fighting for a piece of the pie.

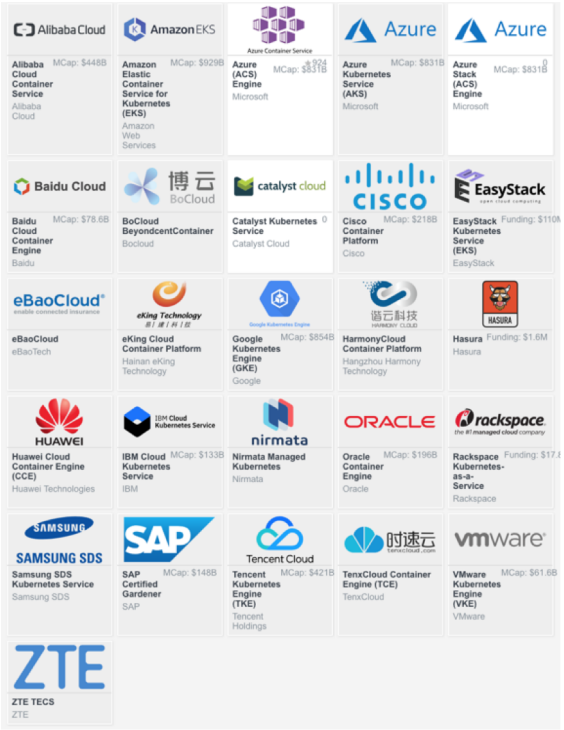
https://www.cloudhealthtech.com/blog/container-wars-are-over-kubernetes-has-won

Kubernetes landscape is crowded

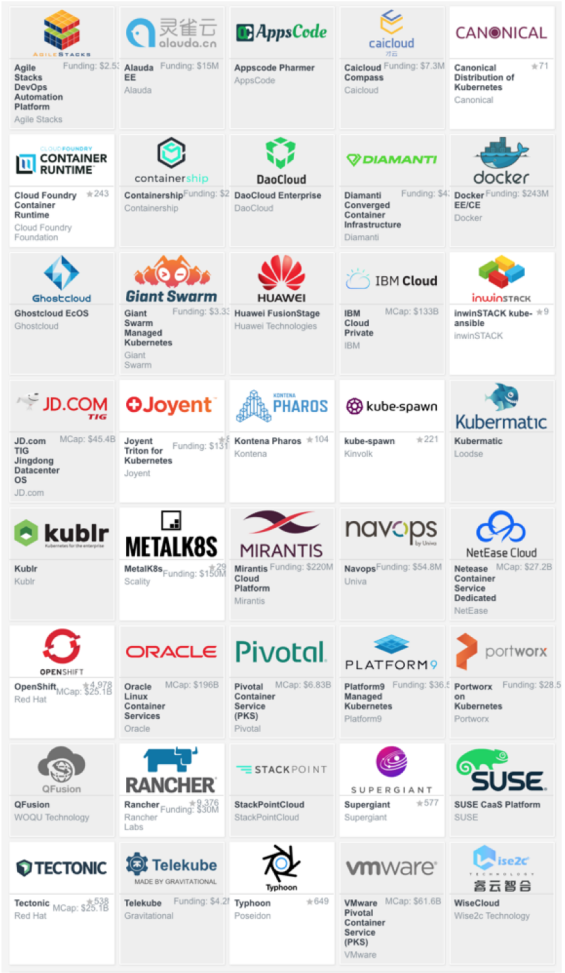
Certified installer



Certified hosted



Certified distributions



https://kubernetes.io/partners/#conformance

Serverless

Serverless Functions

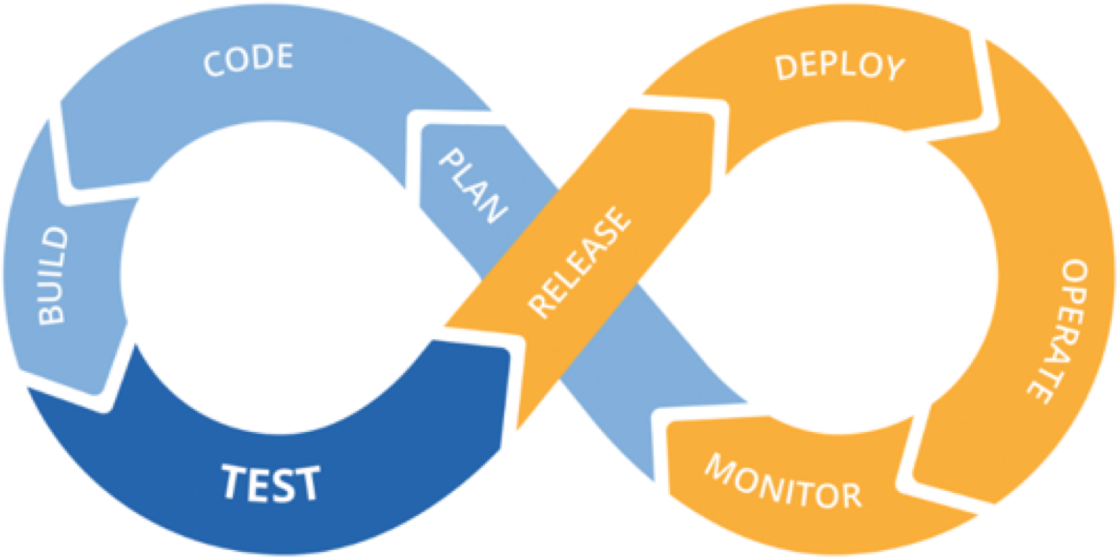
Lambda	Azure Functions	Google Functions	IBM Functions	Fission

Some of the providers
that enable
Serverless Functions

https://headmelted.com/serverless-showdown-4a771ca561d2

CI/CD

Continuous Integration / Continuous Delivery



Don’t leave home without it!

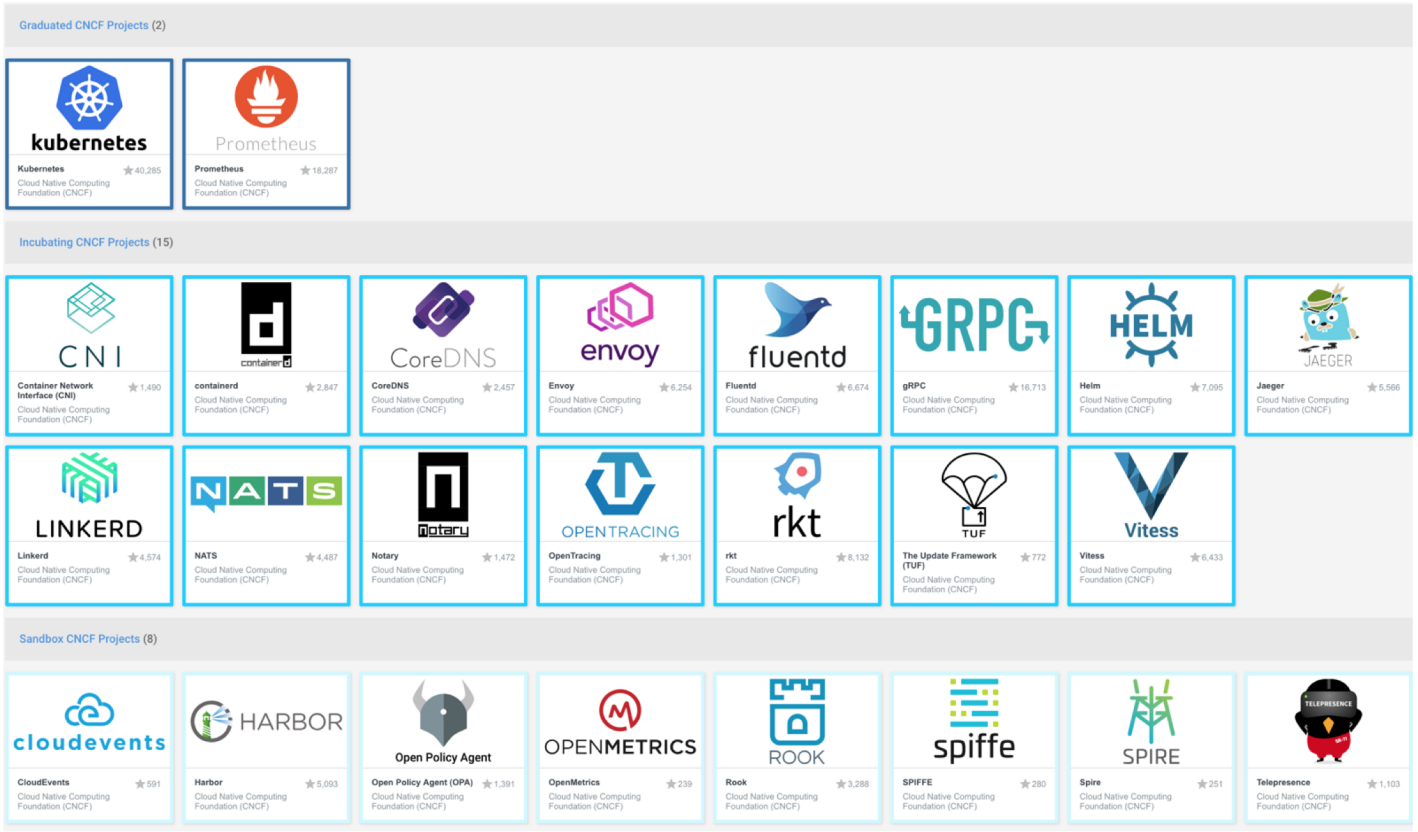
https://blog.gds-gov.tech/that-ci-cd-thing-principles-implementation-tools-aa8e77f9a350

The many different views of CI/CD

https://www.google.com/search?q=CI/CD

What’s next?

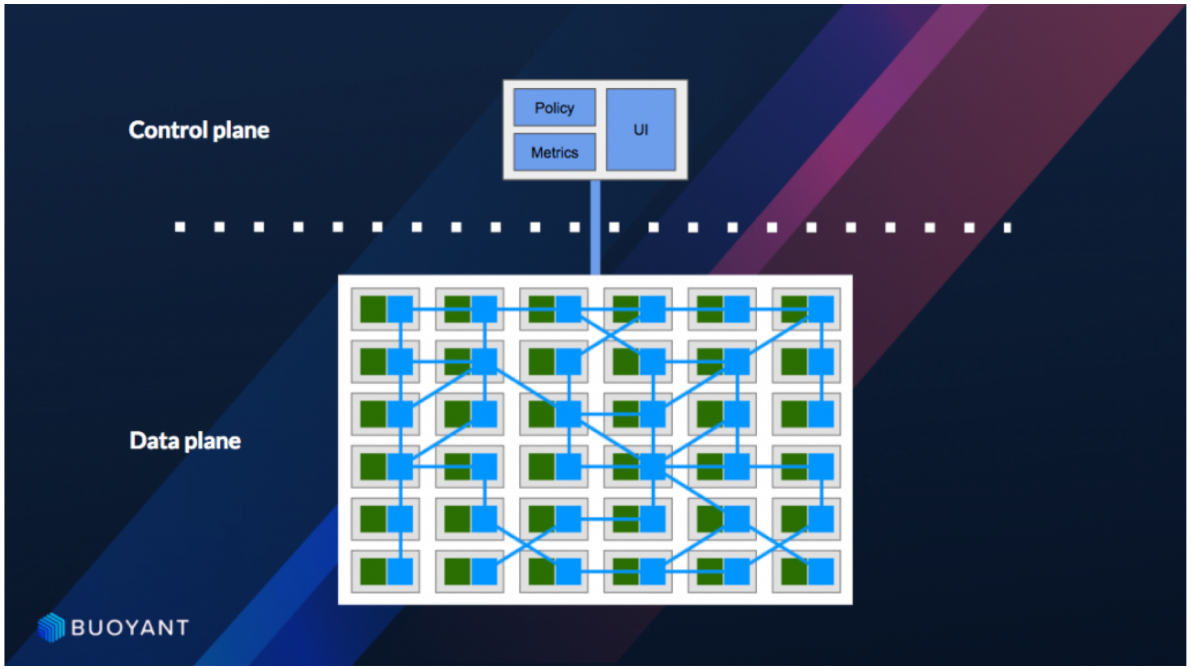
CNCF is helping direct the future



https://landscape.cncf.io/

Service Mesh

Service mesh



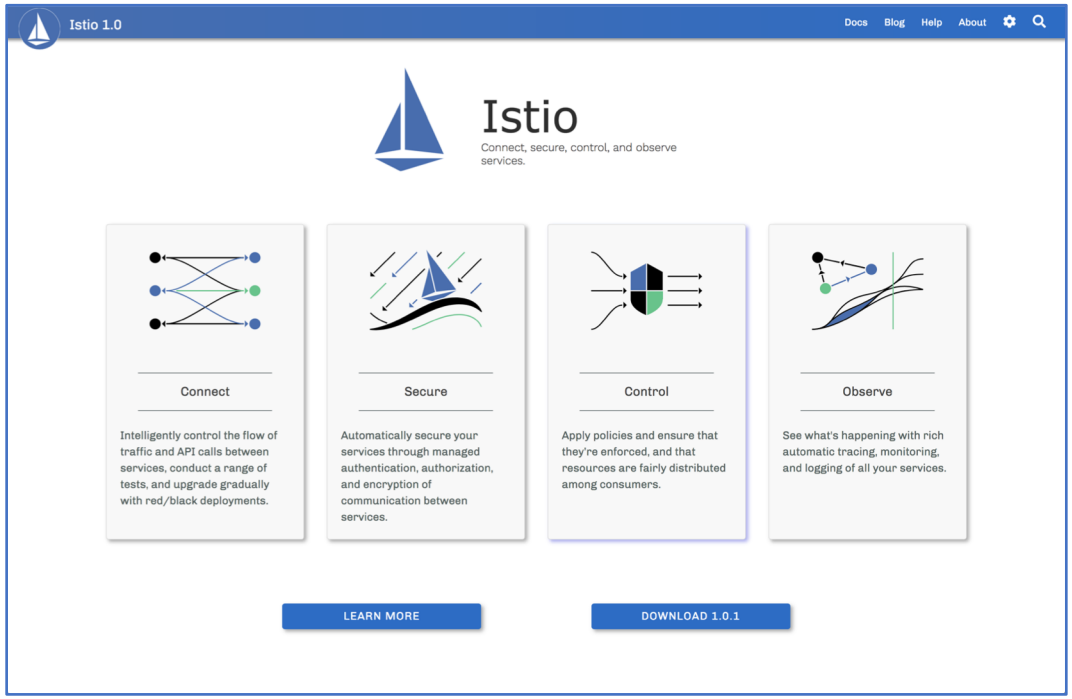
The Common Attributes of a Service Mesh

In the basic architectural diagram, the ***green*** boxes in the data plane represent applications, the ***blue*** squares are service mesh proxies, and the rectangles are application endpoints (a pod, a physical host, etc).

https://thenewstack.io/which-service-mesh-should-i-use/

Istio – Service mesh

Istio provides behavioral insights and operational control over the service mesh as a whole, offering a complete solution to satisfy the diverse requirements of microservice applications.



https://istio.io/docs/concepts/what-is-istio/

Questions / Comments

Thank you

Hands on labs

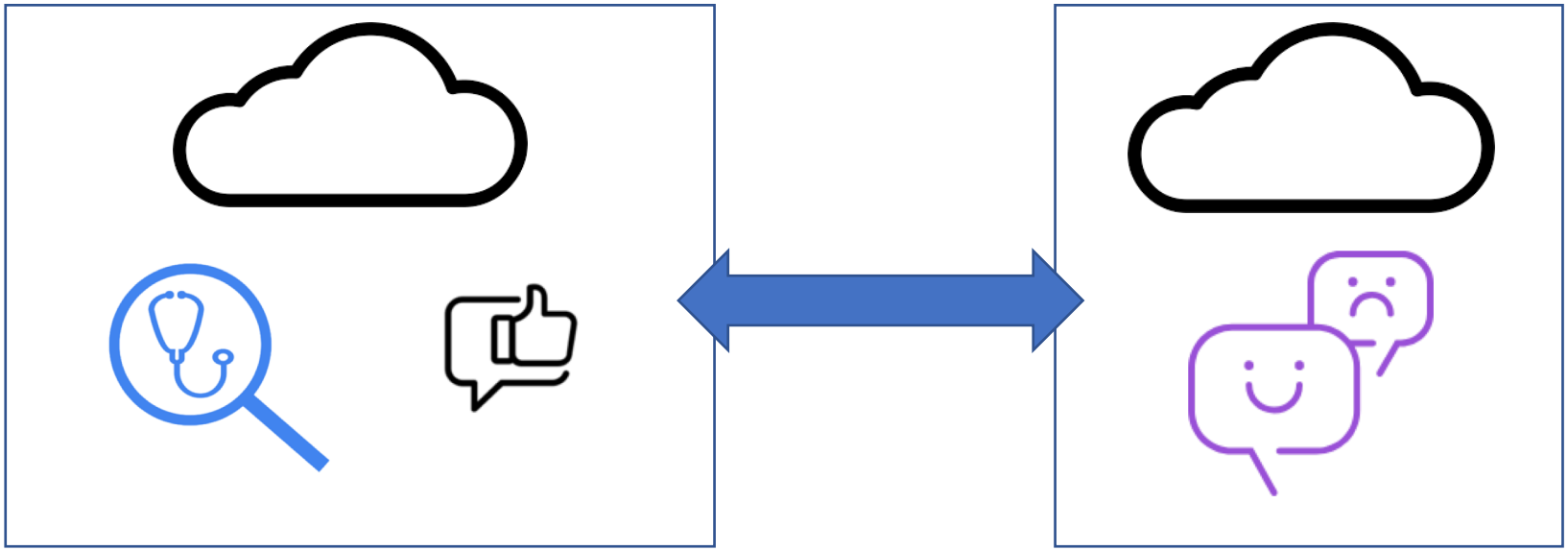
Re-Host (“lift and shift”)

On-premise	Cloud

Business need: Provide national access to application using Cloud provider

IT approach: Re-Host application with maximum portability

Re-Factor



Business need: Enable client feedback with tone analysis and results dashboard

IT approach: Use microservices to deliver feature in conjunction with 3rd party SaaS Tone Analysis