BOREALIS

Enterprise Container Ecosystem (AKS)

HCL 2030 Technology for the Next Decade, Today.

AGENDA

- HCL understanding of Borealis Requirements
- Units of deliverable and Estimates
- Architecture, Design and Deployment of AKS
- Multi – Cluster deployment (HA) and DR deployment
- Monitoring and Security solution

HCL

2030

Fast Forward to the future

Technology for the next Decade, today.

HCL Understanding of Borealis Requirements

Understanding of Requirements

Current Scenario :

1. At present Containerized Application are deployed on AWS (ECS EC2 launch mode).
2. At present CI/CD tools stack is deployed at On-premise.
3. Container registry in AWS.
4. DNS routing in clusters is currently handled by Borealisdigitalstudio.com.
5. Every application is deployed as sub-domain.
6. There is no third party proxy solution deployed on AWS.
7. Borealis has deployed open source projects that provides strategic values to OWASP and application security as whole.
8. Azure Foundation is not ready yet.

Future Requirements :

1. Azure Region : Amsterdam
2. Kubernetes Deployment : AKS .
3. Container registry : ACR
4. Key Vault : Azure Key Vault.
5. Separate cluster for Development/Testing and Staging/Production.
6. In Staging/Production, Application will be deployed in HA and DR.
7. Logging, Monitoring and Security solution for Containers Application.

Scope of work

In scope :

1. Architecture and Design AKS Clusters, ACR, Azure Key Vault, Azure Monitor for Development, Testing, Staging, Production and DR clusters.
2. Build AKS cluster for Development and Testing with minimum requirements for Azure Cloud Foundation. This can be considered to be Developer's Sandbox.
3. Integrate AKS with CI/CD tools and Azure Monitor.
4. Provide Assistance in Application On-boarding (Sample applications).
5. Architect, Design and Build Container Monitoring and Security solution (**Azure Native**).
6. Conduct a POC for Container Monitoring and Security.

Out of scope :

1. Azure Cloud Subscription.
2. Complete / Full scale Azure Foundation.
3. Azure Network connectivity with On-premise Data Center(s).
4. Build Staging/Production/DR clusters.
5. Build and Deploy Monitoring and Security solution leveraging Third-party solutions like SysDig etc
6. Application Migration on AKS.

Project Deliverables

Track 1

- Multi-Cluster/Nodes/Namespace/Pods setup
- Cluster Access Management
- Cluster Configuration Management
- Base Image Lifecycle Management
- Application container Image Life cycle Management (démo with sample applications)

Deliverables

- Architecture & Design document
- Best practices for Day 0 and day 1
- Implementation document
- Build script for AKS, ACR & Key Vault
- Service Catalog (Cloud Services Scripts & Template)
- All the above requirements will be validated along with NFRs viz. Security, Confidentiality, Accessibility, Availability, Scalability, Reliability Usability, Maintainability, Modifiability, Flexibility.

Track 2

- Setting up a Dashboard for Logging, Monitoring and Security
- Error traceability and application runtime performance
- Application Performance Monitoring
- Authentication, Image Scanning, Runtime security in CI/CD and API Gateway

Deliverables

- Build Scripts for Monitoring and Logging systems
- Other Configuration scripts
- Best practices and Guidelines
- Governance documentations
- DevSecOps Pipelines
- Platform Setup & Costing

HCL

2030

Fast Forward to the future

Technology for the next Decade, today.

Unit of Deliverables

Work Packages

HCL AKS Projects Deliverables

Work Package 1 - Planning and Strategy
3 Weeks

- Facilitate 1 kickoff meeting of up to 4-6 hours in duration
- Provide general project overview, provide HCL's schedule of activities and review of Key Dependencies
- Introduce HCL's stakeholders and Borealis stakeholders.
- Review project governance structure, change management process, prerequisites and assumptions, training and knowledge transfer requirements.
- Understanding delivery model & processes, compliance & security requirements and controls fitment, ecosystems like monitoring, logging, storage etc.
- Design document for container registry (Image Signing and Geo-Replication, Perimeter Security, Authorization and RBAC), automated security testing during build, in image registry and runtime.
- Document the Implementation plan (Low level design)

Work Package 2 - Build and Deploy
3 Weeks

- Configure VNET, Security Groups, Routing and Storage
- Configure Azure Firewall (Optional)
- Configure ACR
- Creation of AKS cluster with required configuration, Azure LB, App Gateway and target group setup
- Configure Azure Key Vault
- Assist Borealis team with service creation, and onboarding of workloads
- Configure Azure Monitor and App Insight to collect, analyze, and act on telemetry data from AKS environment.

Work Package 3- Validation and Testing
2 Weeks

- Run sanity tests for AKS, ACR and Azure Key Vault
- Training to Borealis staff
- Handholding and reverse KT
- Submission of documentation

Proof of Concept

Integrated Monitoring & Security solution

1. Presales/Demo SysDig solution for Monitoring and Dashboard
2. Container security configuration and Image security compliance workshop

** If Borealis agrees on SysDig solutions merits, HCL will deploy SysDig in Borealis environment as an additional work package for all environments.

Note - We will submit a detailed SOW

HCL

2030

Fast Forward to the future

Technology for the next Decade, today.

Effort Estimation

High Level Estimation of Effort and Project timeline

Phase - Work Package	Skillset	Location	Level	Duration (Man-weeks)
Complete Project	Project Managemet (Shared)	Offshore	L3	180
Phase 1/WP1	Platform reliability engineer (Tech Arc)	Onsite	L3	3 Weeks
	Azure Security Consultant (Container Security requirements)	Onsite	L3	
Phase 2/WP2	Platform reliability engineer (Tech Arc)	Offshore	L3	3 Weeks
	Platform reliability engineer (AKS)	Offshore	L2	
	Azure Security Consultant (Container Security requirements)	Offshore	L3	
	Platform reliability engineer (Azure Monitor and Azure Insights)	Offshore	L2	
Phase 3/WP3	Platform reliability engineer (Tech Arc)	Onsite	L3	2 weeks
	Platform reliability engineer (Azure Monitor and Azure Insights)	Offshore	L2	
	Azure Security Consultant (Container Security requirements)	Offshore	L3	
Additional Work package - SysDig *	Autonomics SME - SysDig DD, Arch & Design	Onsite	L3	3 Weeks
	SysDig platform implemnation and configuration for Pre-prod and Prod clusters	Offshore	L3	
	Container Security Policies	Offshore	L3	

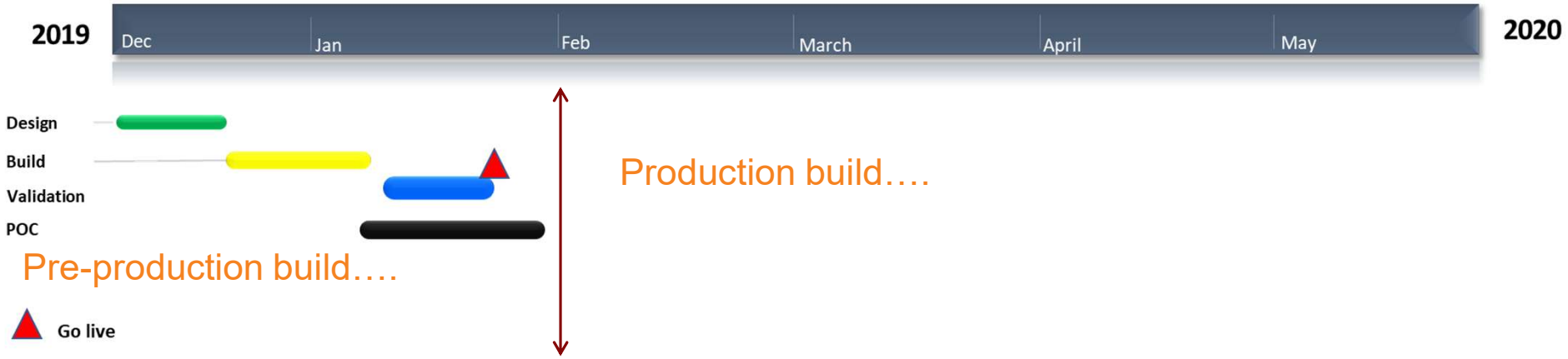
* Additional Work Package – SysDig is optional. The commercial is tentative and can be finalized based on overall requirements for Monitoring and Security (POC).

A Demo will be organized by Presales team during Phase 3.

- Work Package 1 will be delivered Onsite (Belgium). Total duration 3 weeks.
- Work package 2, will be delivered from Offshore (India). Total duration 3 weeks
- Work package 3, will be delivered from Onsite (Belgium).. Total duration 2 weeks
- Payment Milestone – Work package wise

Project Management Approach

- Project timeline –



- Assumptions:
  - The present project does not include build and deploy of Production & DR cluster. Same shall be delivered once Azure Foundation is ready.
  - SOW and commercial for SysDig project will be submitted once POC is successful and Borealis wants to move ahead with SysDig.
  - Project will be delivered in waterfall model, in a hybrid model (mix of onsite and offshore)
  - Leverage **existing** Cloud Project Management service.

HCL

2030

Fast Forward to the future

Technology for the next Decade, today.

AKS Architecture

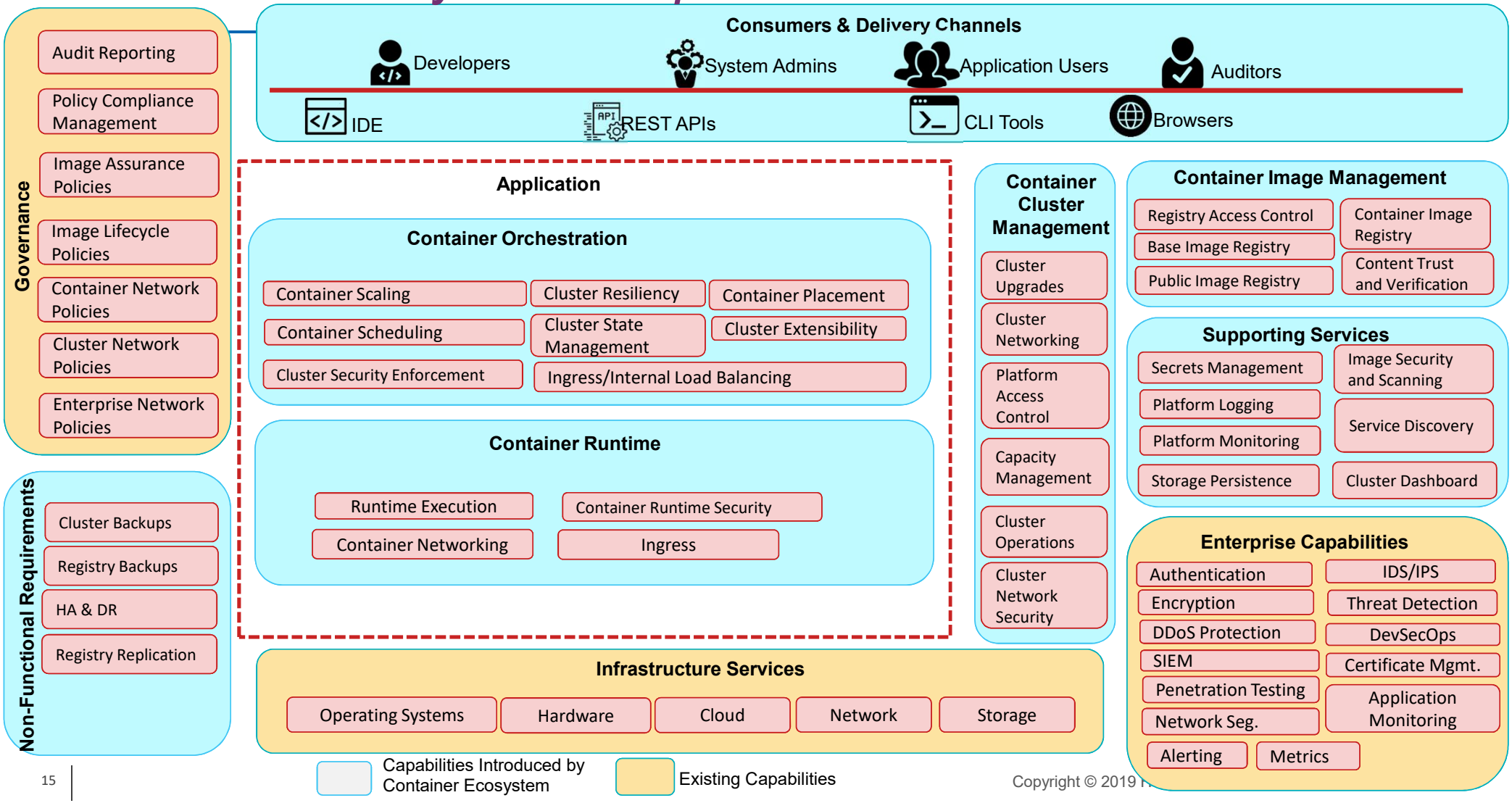
Architecture Vision

- The architecture shall be designed to support the features required by highly available distributed systems, such as auto-scaling, high availability and fully secured.
- Cluster maintains high availability by putting nodes on availability set. Horizontal and Vertical scaling should be a must feature for any Kubernetes Cluster creation.
- The architecture should be built in accordance with existing setup and after due consultation of Borealis Enterprise Architects.
- Cluster segregation, project namespace and cluster DNS management will be done after due consultation of Borealis Application teams.
- The architecture shall be put in place after analysis of Cloud provider (Azure) recommendation and best practice for Azure kubernetes cluster creation and deployment. It shall also cover integration with Borealis tools required to cover ITSM, APM, DevSecOps etc.
- Architecture should addresses security at multiple levels: cluster, application and network. Security controls like TLS, RBAC, NSG, Network policy etc. should be used to show the capability of built system

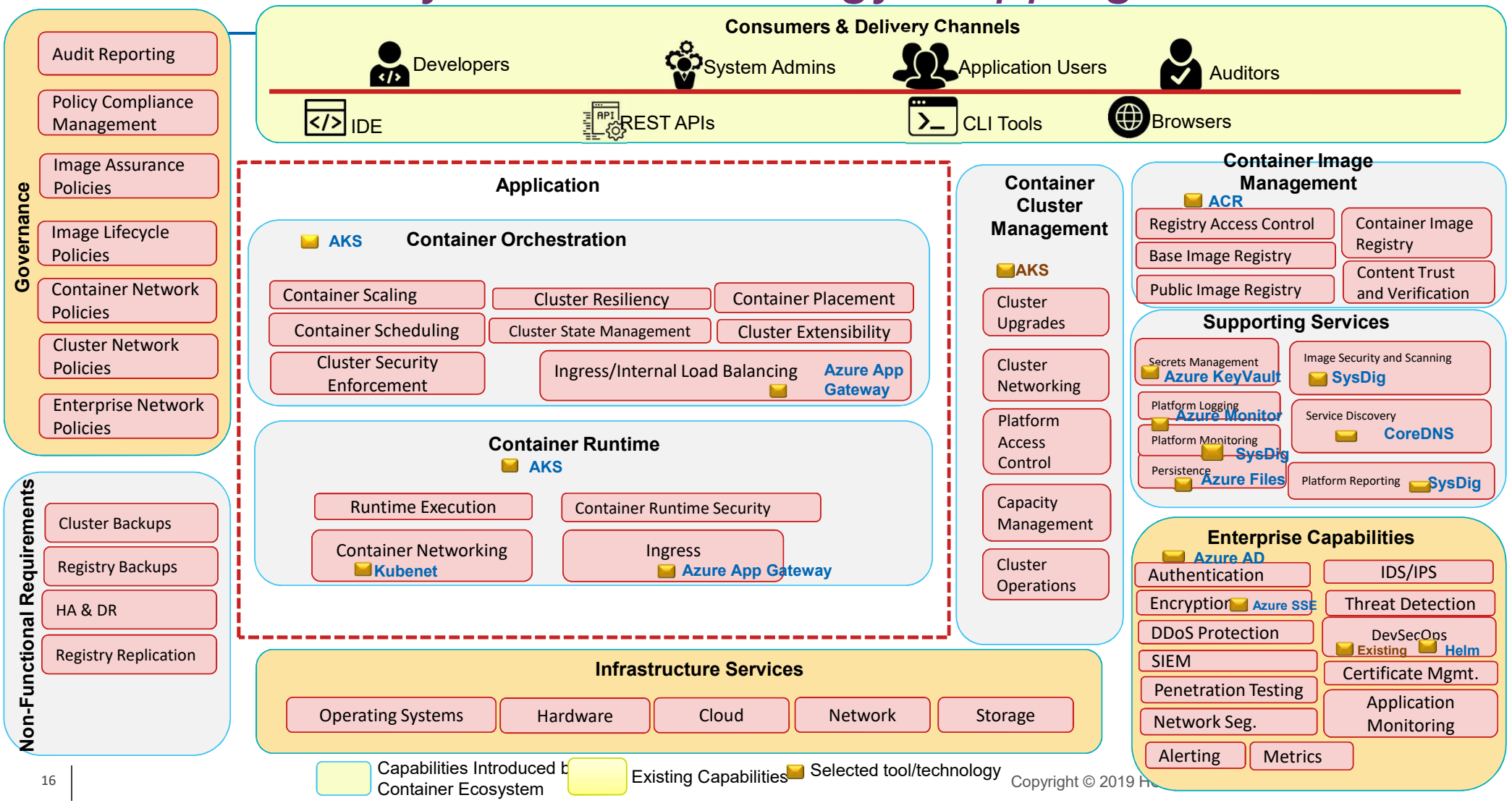
Container Ecosystem Capabilities Overview

S.No.	Domain	Capability	Description	Deliverables
1.	Container Runtime	Runtime Execution	Execution of containers and management of local copy of container images on a node	HCL will consult Borealis on best practices. HCL will build the AKS clusters
2.	Container Runtime	Container Networking	Inter Networking of running containers within the platform through which individual containers can be reached.	
3.	Container Orchestration	Cluster Scaling	Ability to Increase/Decrease number of running containers for each application based on CPU/Memory resource consumption and to Increase/Decrease number of running VM/Nodes on which containers run based on no. of containers running or waiting to be started	Platform capability. HCL will build the AKS clusters
4.	Container Orchestration	Cluster Resiliency	Ability to keep track of running containers and in-case of containers killed or failed, re-creation of those containers.	
5.	Container Orchestration	Ingress	The load balancing mechanism to direct traffic to selected containers based on FQDN and/or paths.	HCL will consult Borealis on best practices. HCL will build the AKS clusters
6.	Supporting Services	Secrets Management	Capabilities to securely store and utilize (by the containers) the secret information such as Passwords, Access Keys and Security Certificates.	
7	Supporting Services	Storage Persistence	Capabilities to use the storage by containers which are not attached to local VM/Node and can be mounted based on requests to provide persist datastore.	

Container Ecosystem Capabilities Details



Container Ecosystem- Technology Mapping



HCL

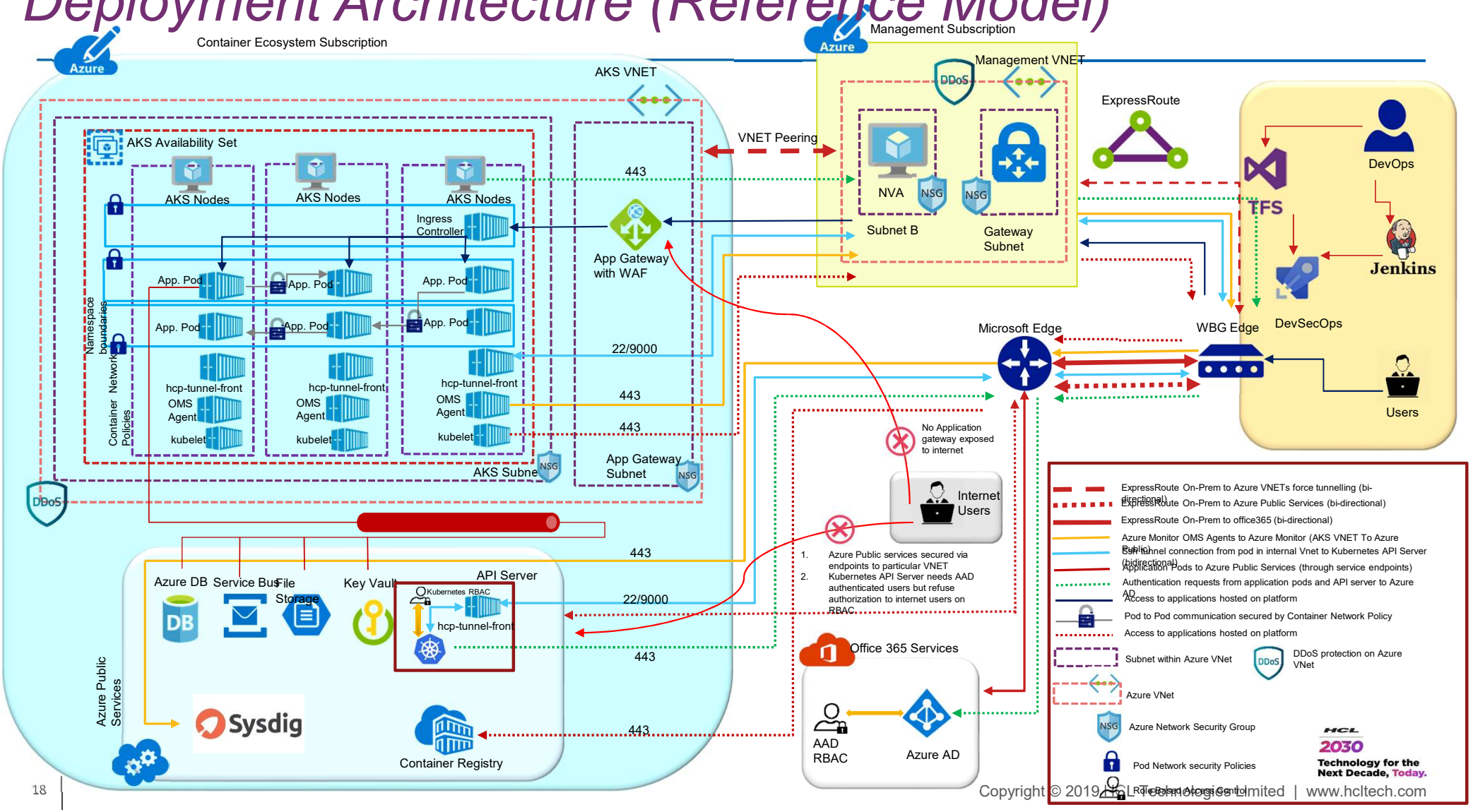
2030

Fast Forward to the future

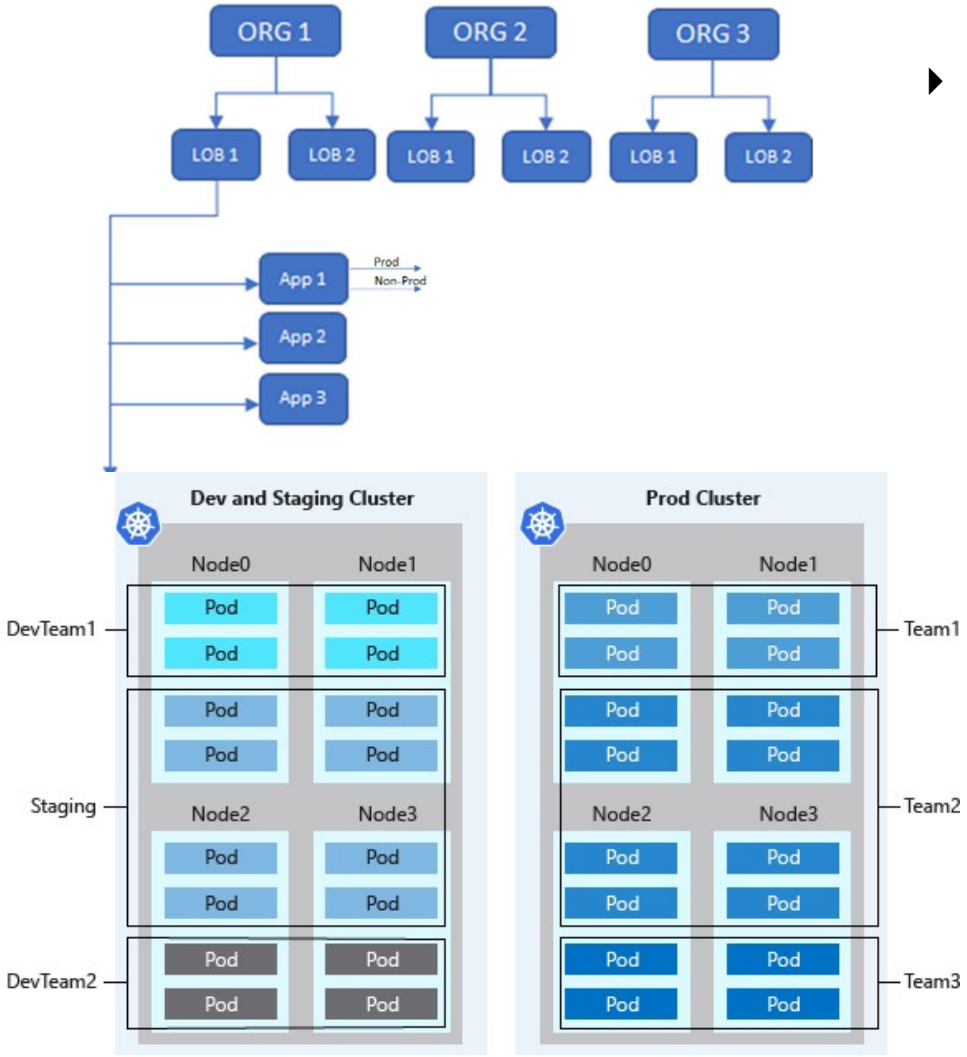
Technology for the next Decade, today.

Deployment Architecture

Deployment Architecture (Reference Model)



Cluster Segregation



- Borealis has different requirements of segregating their cluster based on different criteria. These can be classified based on
  - Organization
  - LOB
  - Application Type
  - Environment

Cluster segregation must be planned based on Borealis existing hierarchy structure.

it is a recommended practice to have separate cluster for Prod and Non-prod environments.

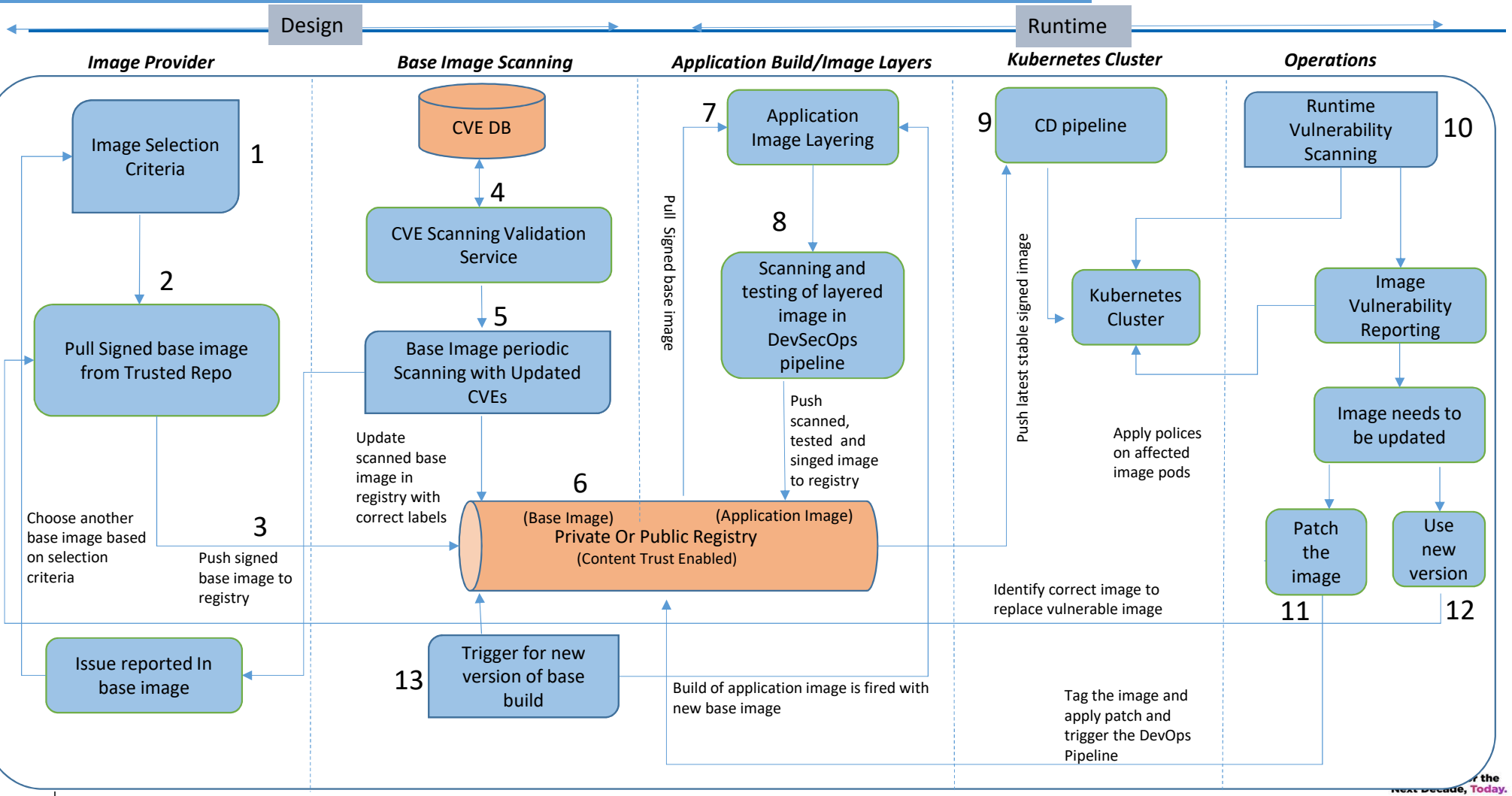
Cluster Networking

- Selection of Ingress Controller and Container CNI. Azure CNI is preferred.
- Exposing Kubernetes workload : Internal Load Balancer (Azure Application Gateway Ingress Controller)
- Pod level failure is addressed by Kubernetes itself, where it spins another pod, if one is not responding. By Default, a CNI plugin enables maximum 30 pods in a subnet. In a /24 CIDR, there can be maximum of 8 nodes.
- Any scaling expectation should be planned by increasing the range of the subnet.
- For Non-Prod environment, organization can live with cluster in a region. HA and DR Planning is a must in a Prod environment.
- To protect from region failure, deploy the application into multiple AKS clusters across different regions and traffic between it should be managed by load balancer.
- CI CD pipeline should ensure deployment of latest images on both clusters. Container registry should also be deployed in multiple region with sync enabled.

Capabilities/Features	Kubenet	Azure CNI	Calico
IP Range assigned to Pods/Containers	Virtual and visible only to Kubernetes Cluster	VNET IP Addresses which are addressable from Azure connected network.	Virtual and visible only to Kubernetes Cluster
Managed Kubernetes Provider specific Support	Default in AKS, not supported in EKS	Supported in AKS by Microsoft	Not Supported for IPAM but for NetworkPolicies from Community only.
Pod Limitations per Node	Default – 110 Max. 250	Default- 30 Max – 250	Not restricted
NetworkPolicy Support	Through kube-router, only community support	Native Azure and Calico	Calico NetworkPolicy

Capabilities/Features	ingress-nginx	istio ingress	Azure App gateway Ingress
Project Release Status (Status of the technology if in GA/Beta/Preview)	General Availability	General Availability	Beta
Project Licensing (Licensing Models)	Community (part of Kubernetes), Commercial version available	Community	Community (By Microsoft)
Supported Protocols (Protocols supported for Ingress Controller)	http,https,tcp (separate lb),udp,grpc	tcp,http,https,grpc	http,https
Base Technology (reverse-proxy/gateway which controller uses for underlying routing)	nginx	envoy	Azure App Gateway
Routing Support (Type of Routings for backend services)	host, path	host, user (Istio routing takes care of path based routing)	Host, path
Kubernetes Scope (Ability to use same Ingress across the namespace or dedicated ingress for each namespace)	cross-namespace	cross-namespace	Single namespace
Dashboard for Metrics	Prometheus and Grafana	Prometheus and Grafana , tracing in jaeger or zipkin UI	Azure Monitor
Support Availability for Product Vendor	Yes, for Commercial version	Community	Community

Image Lifecycle Management



Base image build update

Platform Security Solutions POV

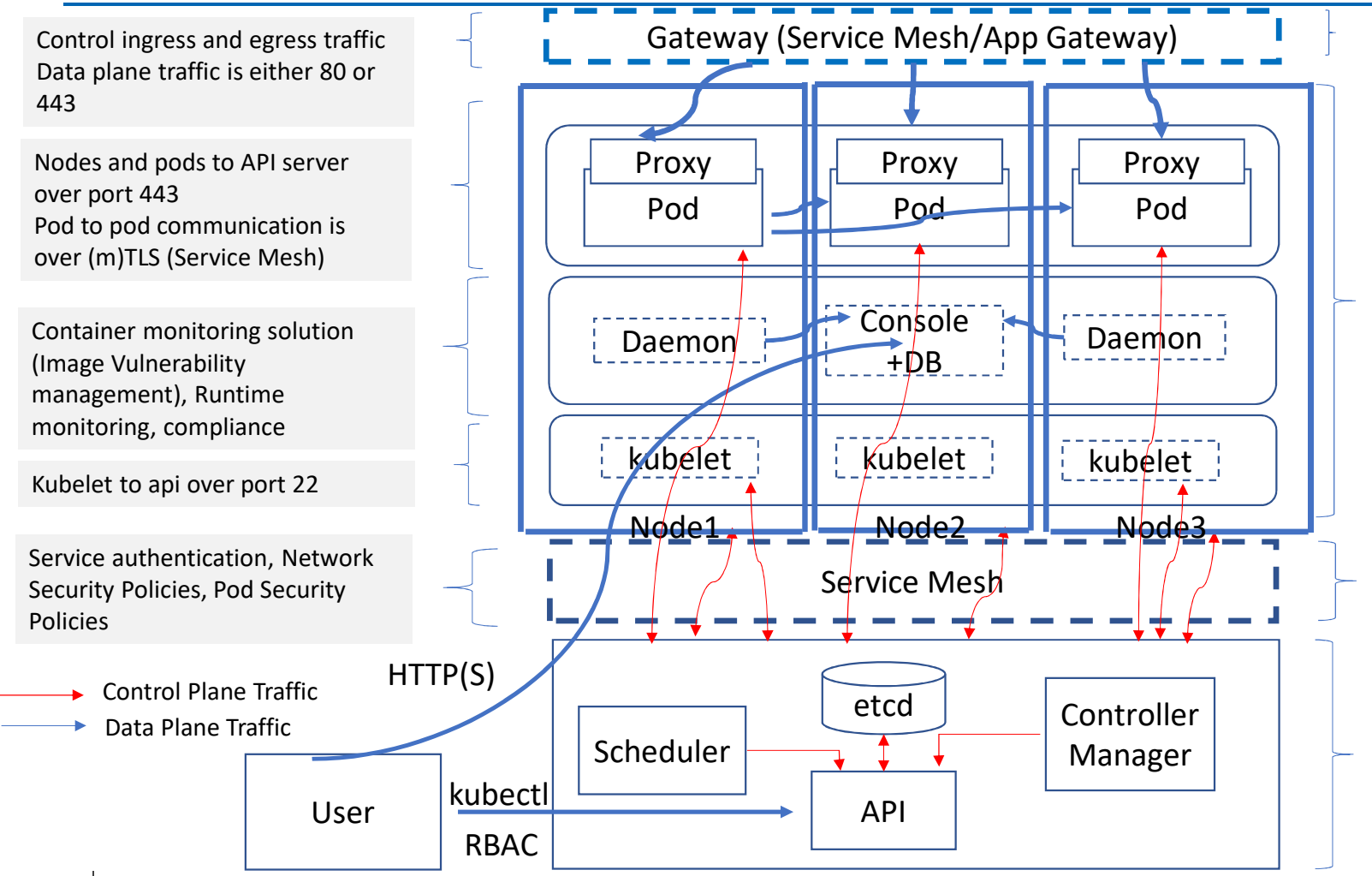
Control ingress and egress traffic
Data plane traffic is either 80 or 443

Nodes and pods to API server over port 443
Pod to pod communication is over (m)TLS (Service Mesh)

Container monitoring solution (Image Vulnerability management), Runtime monitoring, compliance

Kubelet to api over port 22

Service authentication, Network Security Policies, Pod Security Policies



Customer managed

Security for Worker nodes are managed by Azure

Customer to manage the pod definitions and the security console

Customer Managed

Security for Master Nodes are managed by Azure

HCL 2030 Technology for the Next Decade, Today.

Container Security Threat Vectors Analysis

#	Threat	Control
1	Developers commit code to master branch without approval	Developers commit code to master branch without approval
2	Git hub login accounts are either hijacked or compromised	User authentication to GIT hub is tied to multi-factor authentication
3	Build server fetches dependencies and nested dependencies which are outdated and vulnerable	Third party tools can integrate with the Git systems and show vulnerabilities and remediation steps to developers to fix open source vulnerabilities
4	Public images containing vulnerabilities are consumed from registries like Docker Hub	Images needs to be curated and managed in a private registry. Publishers must digitally sign the images before pushing the image to the registry.
5	Vulnerabilities in custom application code	SAST scan can be integrated into the build pipeline which will scan vulnerabilities at code level
6	Vulnerabilities in images	Image scanning can be built into the pipeline which will scan vulnerabilities with images
7	Vulnerabilities in runtime platform	DAST scans can be built into the pipeline for scanning applications in a developer environment
8	Kubernetes secrets are base 64 encoded and are placed as environment variables for applications to consume	A vault can be used to store secrets and pods can connect to the vault to download the secrets at runtime
9	Excessive developer privileges on K8s orchestration platform	Developers will have access to subscriptions in developer environment but will not have any access to the production environment subscriptions. The developer environment are basic subscriptions which are not connected back to the on premise network. The production network is connected back to the on premise environment.

Container Security Threat Vectors Analysis

#	Threat	Control
10	Subscribers to a topic are not authorized against access control policies on message oriented middleware (MOM) systems like RabbitMQ, Kafka etc.	Third party tools can build relationships between publishers and subscribers by introspecting container communication across MOM
11	East west traffic between pods are not authenticated or authorized	Each service communicating needs to be authenticated and authorized. A service principal needs to created which is bound to a service and is authenticated and authorized by the target service
12	If a pod is compromised it should not perform privilege escalation and affect the worker node or other services running within the container	The pods deployed on the K8s should not run as root and pod security polices can be configured to prevent privilege escalation attacks. Nano-segmentation can be done which will allow container based communication between services which need to talk to each other while restricting communication with others
13	K8s exposes an external api, which are consumed by untrusted applications (outside the trusted boundary of the application)	API gateways can be used to reverse proxy connections to the api. For authentication mutual TLS can be used. A better approach would be to use user authorization grant.
14	K8s exposes an external web service which is consumed by partners /vendors /customers/consumers	Depending on the risk profile of the application external controls can be placed on a DMZ containing DDoS, WAF controls

HCL

2030

Fast Forward to the future

Technology for the next Decade, today.

Monitoring and Security Solution

DevSecOps Accelerator – Infra As A Code Convergence

CI/CD

Azure DevOps 2019

Jenkins

Configuration Management

ANSIBLE

Terraform

Orchestrator Tools

ARM template

Build, Test and Integrate applications

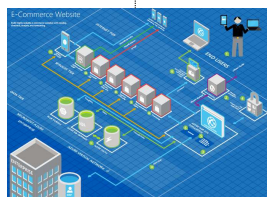
Performance, Smoke and DR testing

Configure and setup an existing machine

Provision Infrastructure and Service based on a configuration template with Policy (Including Security , RBAC, Secret management , Audit)

code-artefact-765

Cloud Native Container



Monitoring & Security Tools Integration

Sysdig

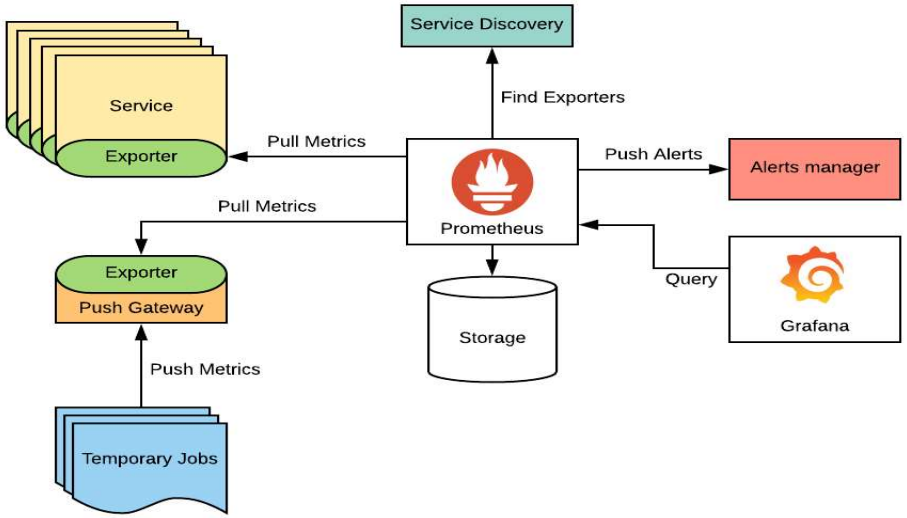
- Web/Application/Database
- Additional Package/software
- GCP/Azure/AWS/VMware/ Container Image

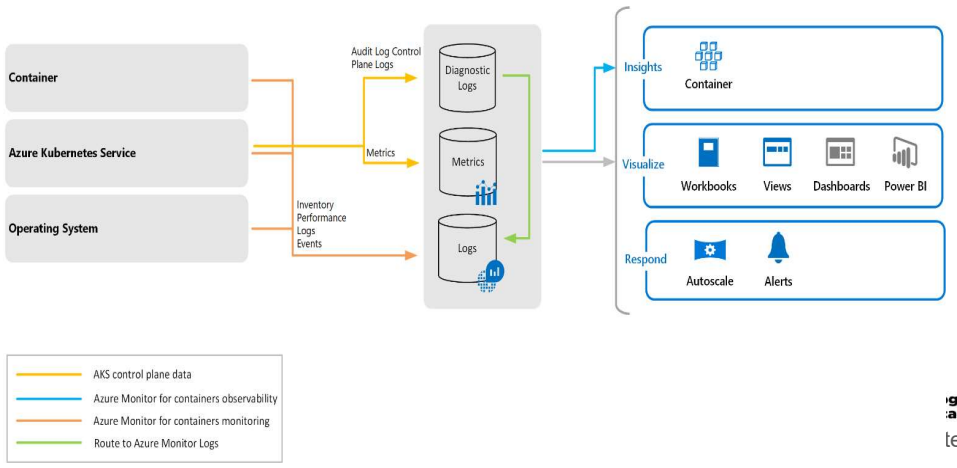
Microsoft Azure

HCL 2030 Technology for the Next Decade, Today.

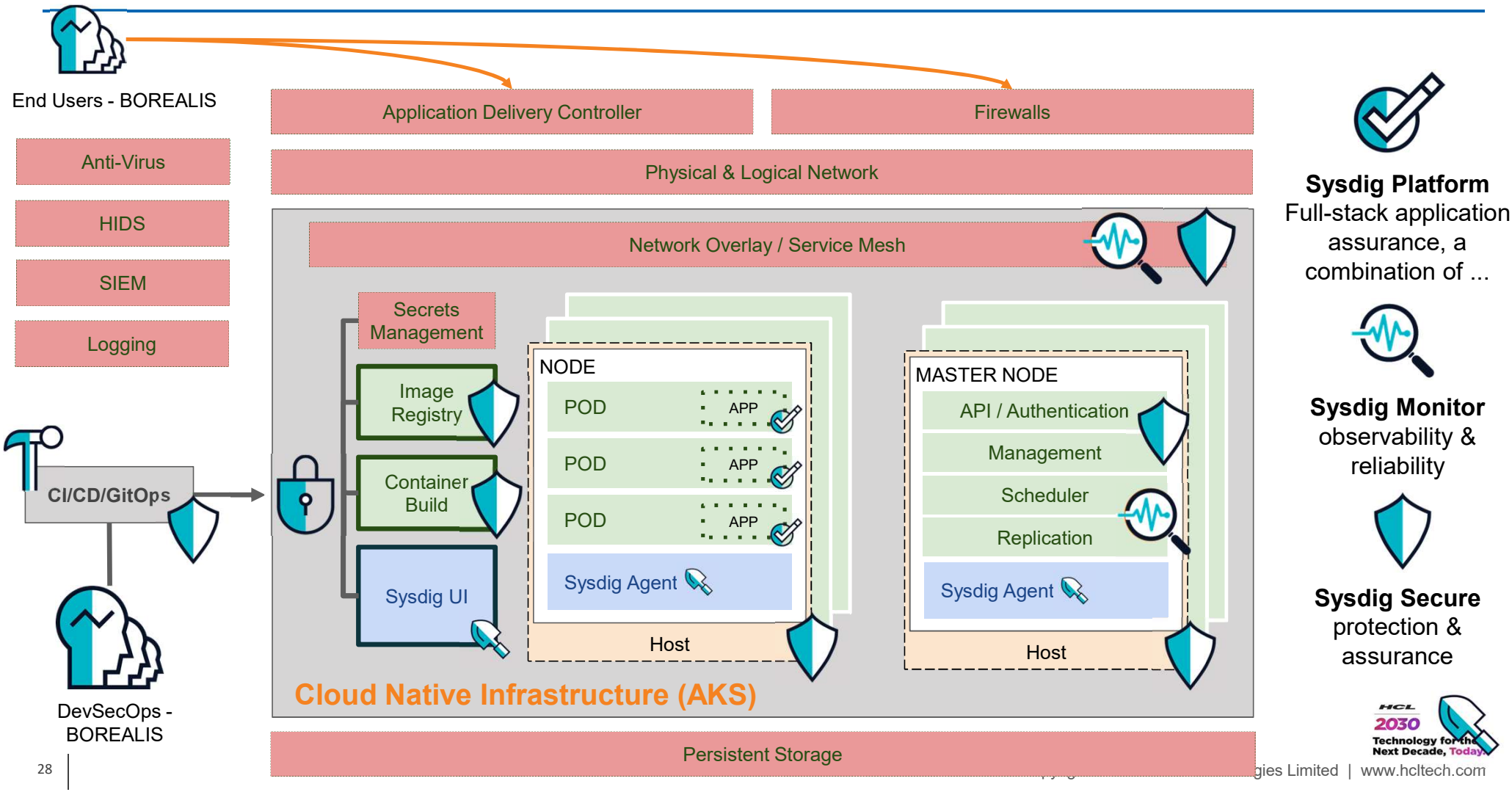
Logging and Monitoring

- Logging activities for platform can be delivered using Azure Monitor and Log Analytics, however in order to provide demonstration of same metrics across multiple clusters, **Prometheus** is recommended to capture the metrics data.
- HCL recommends SysDig which is based on Prometheus and can meet Logging, Monitoring and Container security requirements.
- Following diagram shows how Azure monitor captures events and logs and visualization options in the Azure Portal.





Complementary For Increased Assurance



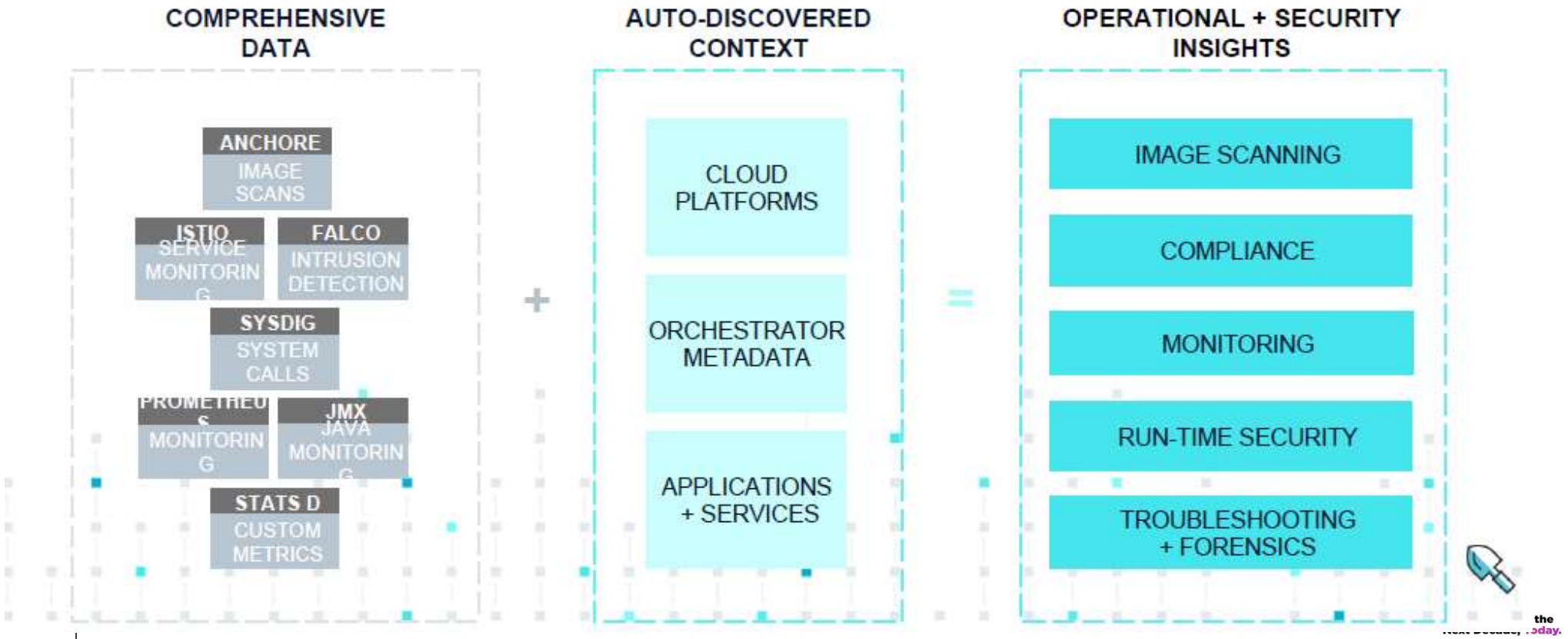
Sysdig Platform
Full-stack application assurance, a combination of ...

Sysdig Monitor
observability & reliability

Sysdig Secure
protection & assurance

SysDig solution

- Single platform which will provide Container Monitoring, Logging, Security and Dashboard across Azure, AWS and On-premise Container Platform Deployment



SysDig Features & Capabilities

Kubernetes insights

- Heavy focus on Kubernetes and ecosystem integrations
- Enrich all metrics + events + visualizations with K8s context
- Monitor kube-components
- Insights for service owners and cluster operators

Ecosystem Integration

- Service mesh visibility: Istio, AWS App Mesh, Linkerd
- New cloud provider offerings: e.g., GKE On-Prem
- Runtime engines: CRI-O containerD
- Operator frameworks: OpenShift
- OpenMetrics (Prometheus)

Enterprise-grade features

- Scale to millions of metrics per second
- Retain data longer for historical analysis
- Multi-cloud: monitor any combination of clouds
- Control, isolate and secure service and data access

Trace-driven troubleshooting

- Capture all system call activity upon alert
- Accelerate response with precise record of system & container data
- Find and fix problems *minus* the overhead of log analysis

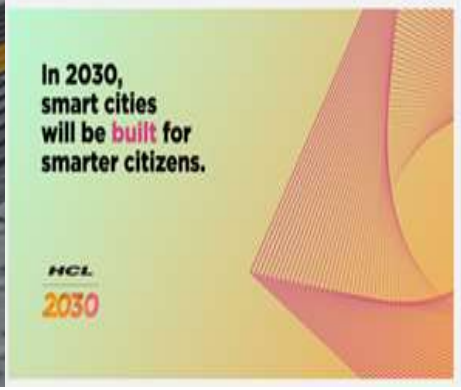

The World is on
the fast-lane
to 2030,
but we need to
act now.


In 2030,
smart cities
will be built for
smarter citizens.
HCL
2030


In 2030,
big data will guide
big decisions.
HCL
2030


In 2030,
equitable growth and
exponential innovation
will converge.
HCL
2030


In 2030,
cryptocurrency
will cease to
be cryptic.
HCL
2030


In 2030,
smart cities
will be built for
smarter citizens.
HCL
2030


In 2030,
IoT will stand
for the
internet of trust.
HCL
2030