A Hybrid Integration Architecture update

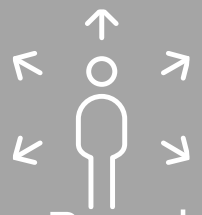
Dominic Storey
IBM

November 6th
Session: JK

Please note

- IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM’s sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

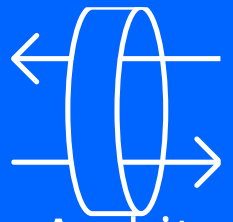
- Rethinking your approach to integration



- People & Process

- How can we improve development agility in order to accelerate innovation?

- **Development Agility**



- Architecture

- How can we improve build independence and increase production velocity?

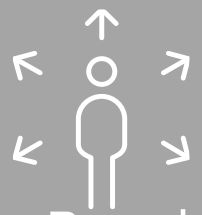
- **Deployment Agility**

- Technology

- How can we improve our ability to deliver dynamic scalability and inherent resilience?

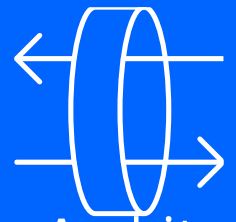
- **Operational Agility**

- IBM can help you modernize with Agile Integration….
- …to achieve development, deployment, and operational agility



People & Process

- Decentralized ownership
- Empowering teams
- Agile methods



Architecture

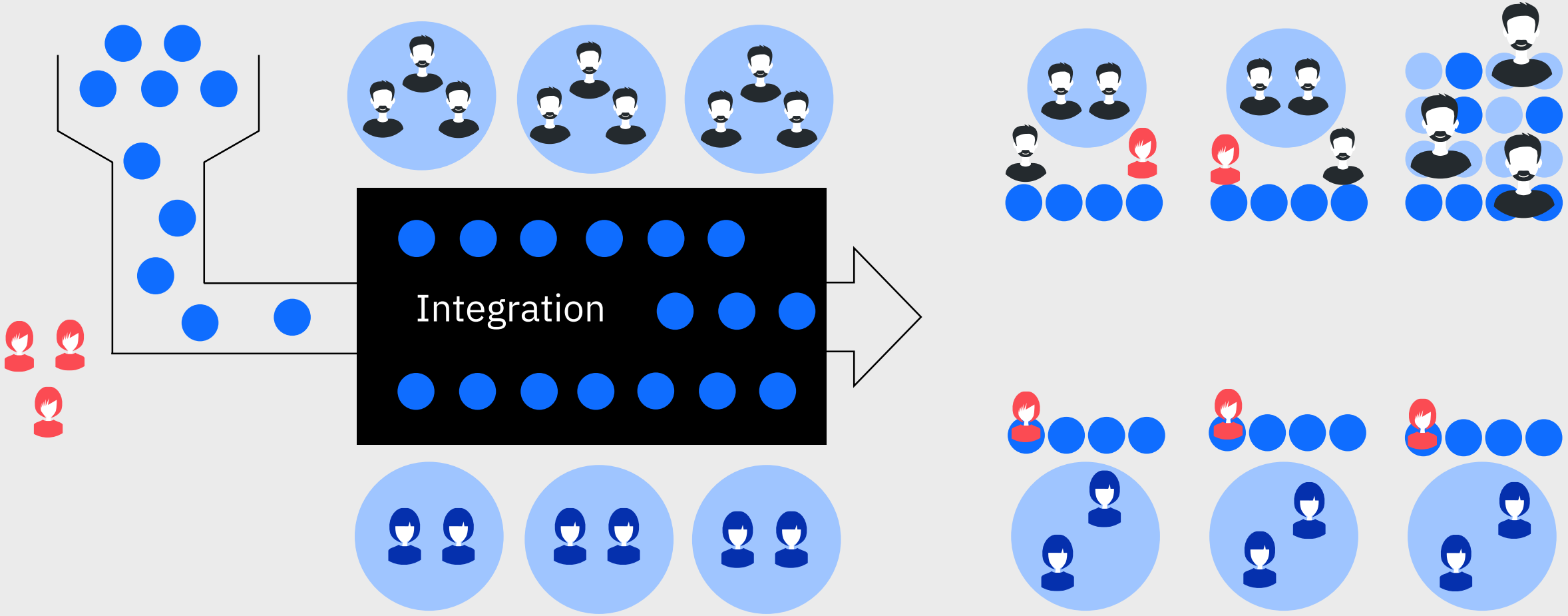
- Fine-grained deployment
- API led
- Event-driven
- Microservices aligned

Technology

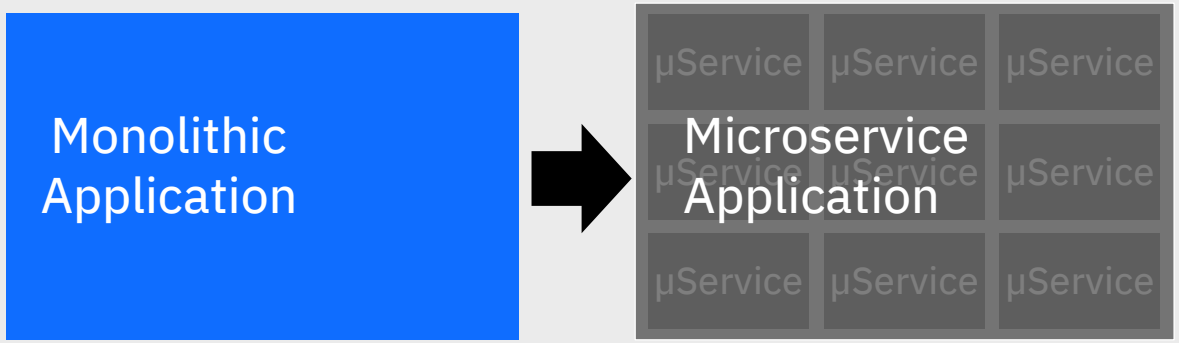
- Cloud-native infrastructure
- Essential integration capabilities
- Unified security, governance, and operations.

People & Process

Decentralized Ownership



Microservices typifies the benefits sought from modern architectural techniques



Agility

Faster iteration cycles, bounded contexts, autonomous teams

Scalability

Elastic scalability, workload orchestration, cloud infrastructure

Resilience

Minimized dependencies, discrete failover, fail fast, start fast

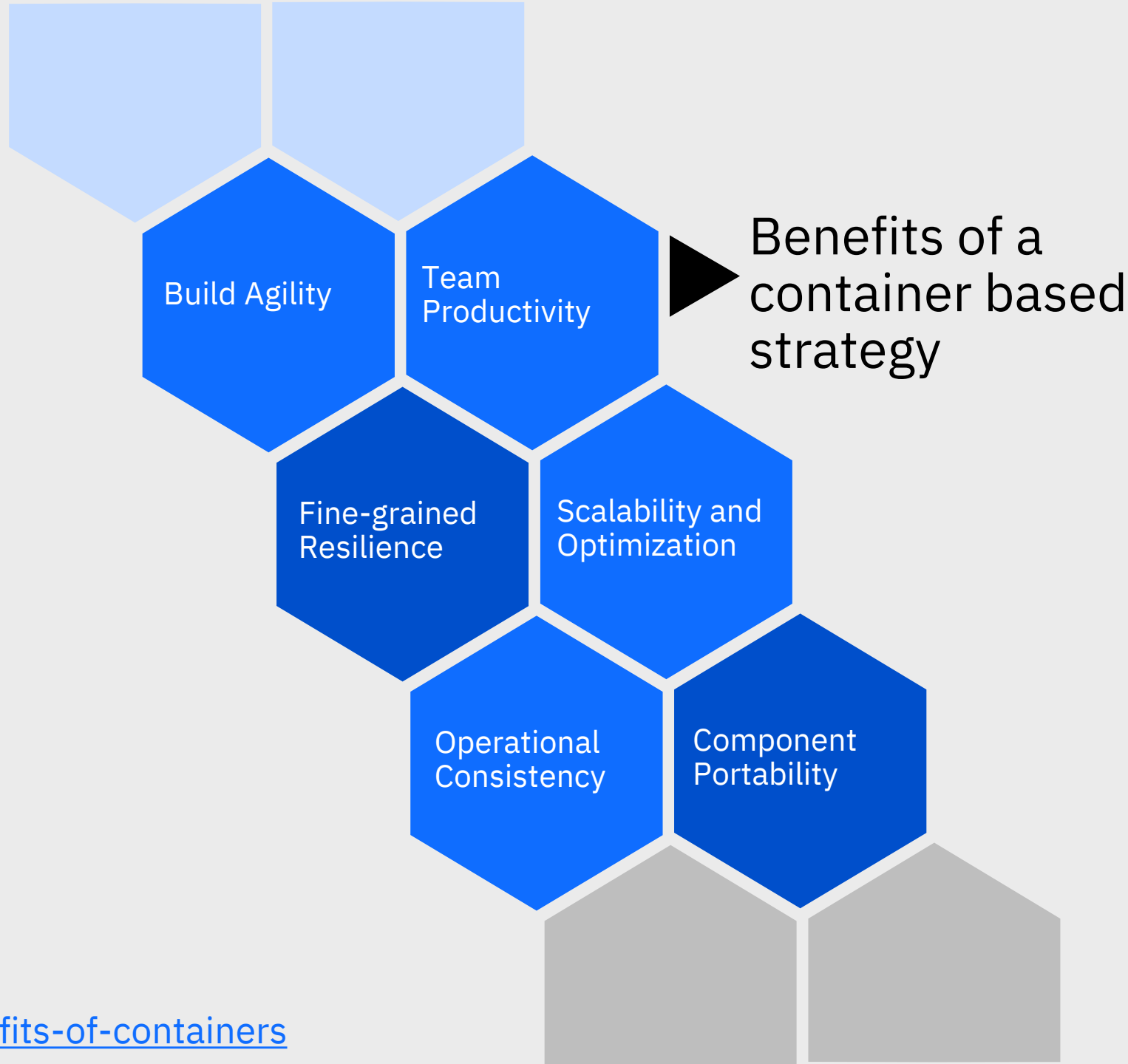
However, microservices is just one of architecture and design influences changing the way we think about building components.
API led, microservices, cloud-native, event sourced...the list continues

Technology

Move to cloud is much more than re-platforming.

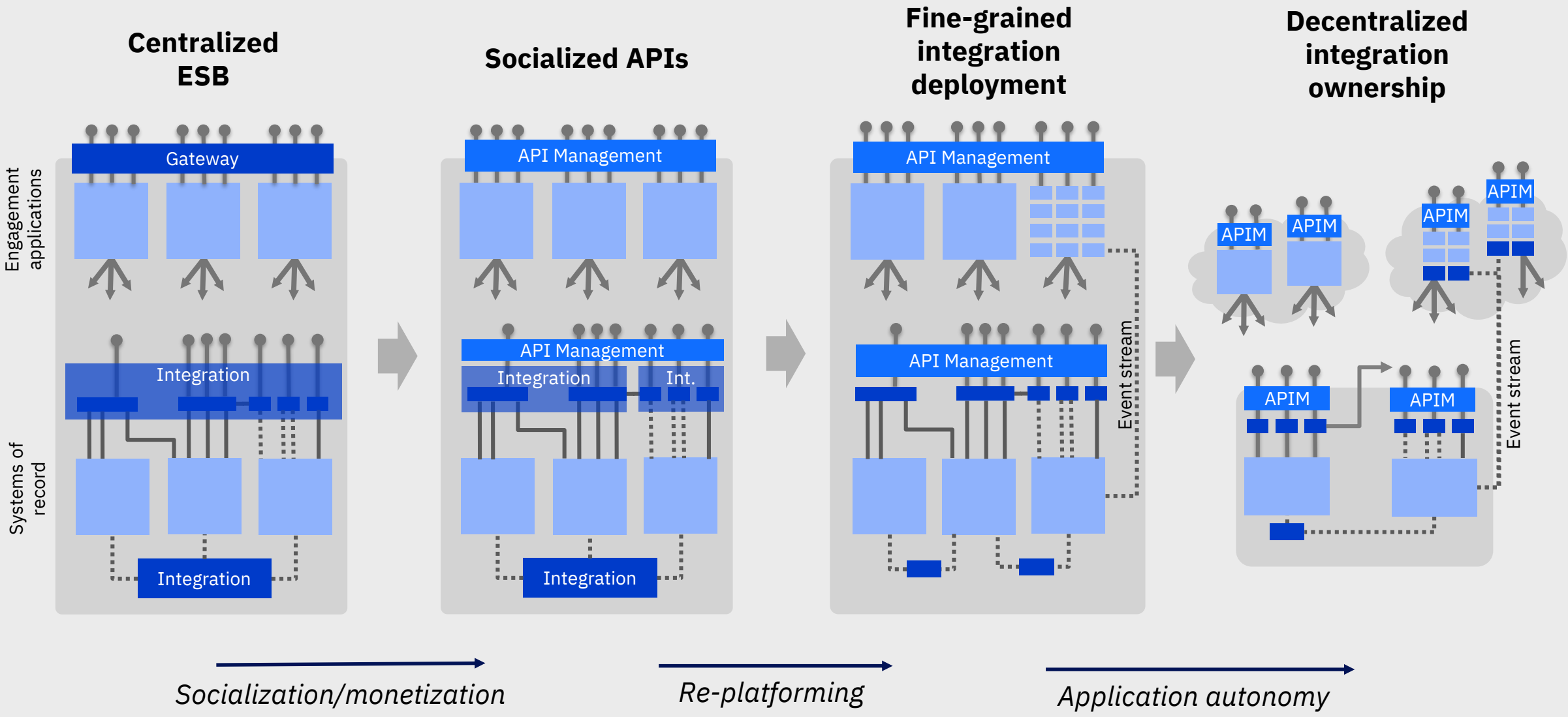
Containers, used in a cloud-native style are part of an evolving story.

Lift and shift will not bring same benefits



https://developer.ibm.com/series/benefits-of-containers

Evolution to **agile integration** – detail view



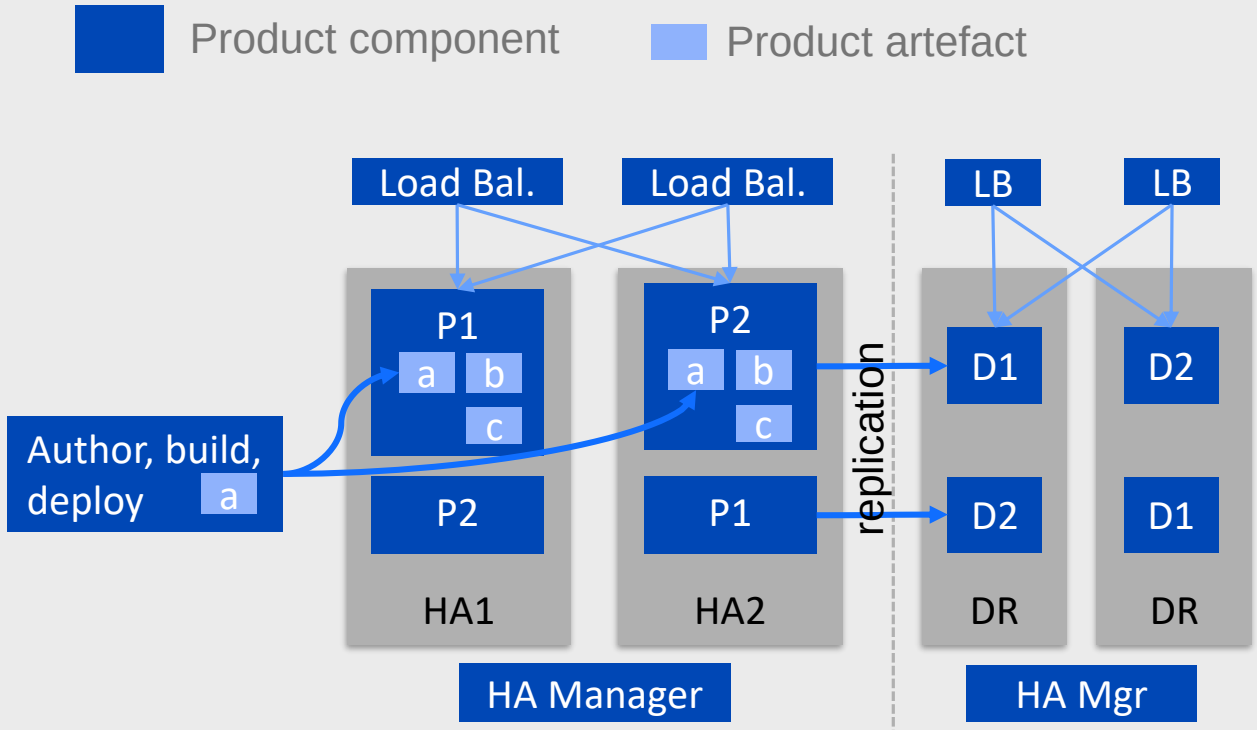
Webinars http://ibm.biz/agile-integration-webcasts

eBook http://ibm.biz/agile-integration-ebook

Traditional vs containerized cloud native

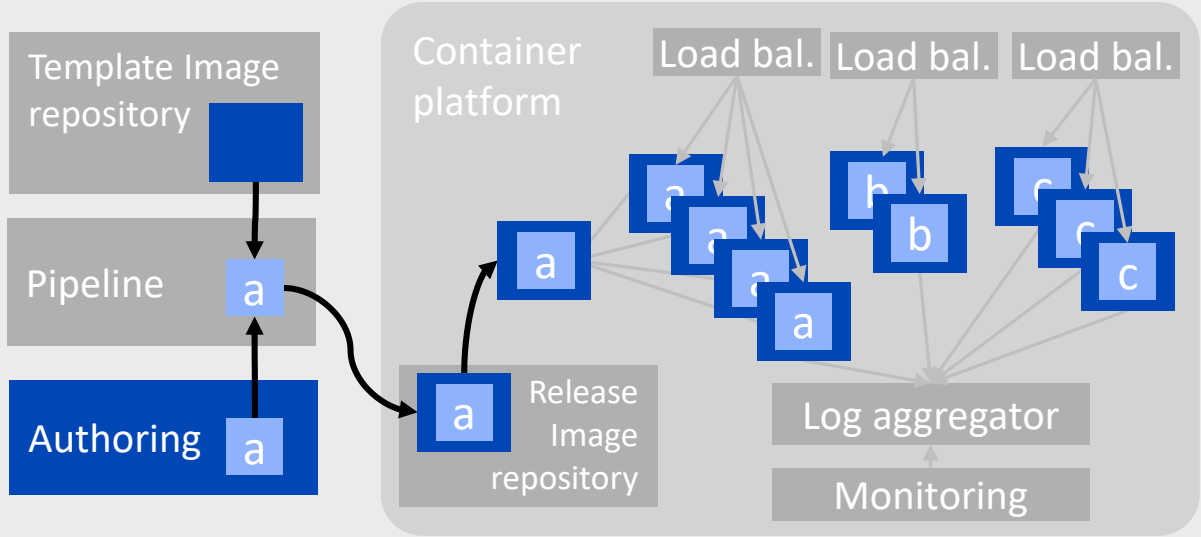
Traditional

Dedicated HA pairs
Scaling manual and vertical
Defined nodes
Explicit install and configure
Explicit cold/warm HA & DR
Peak CPU licensing
Dedicated OS instances/HW
Deploy to running shared servers
Replication across DCs
Administer live shared servers
Code deployed to shared servers

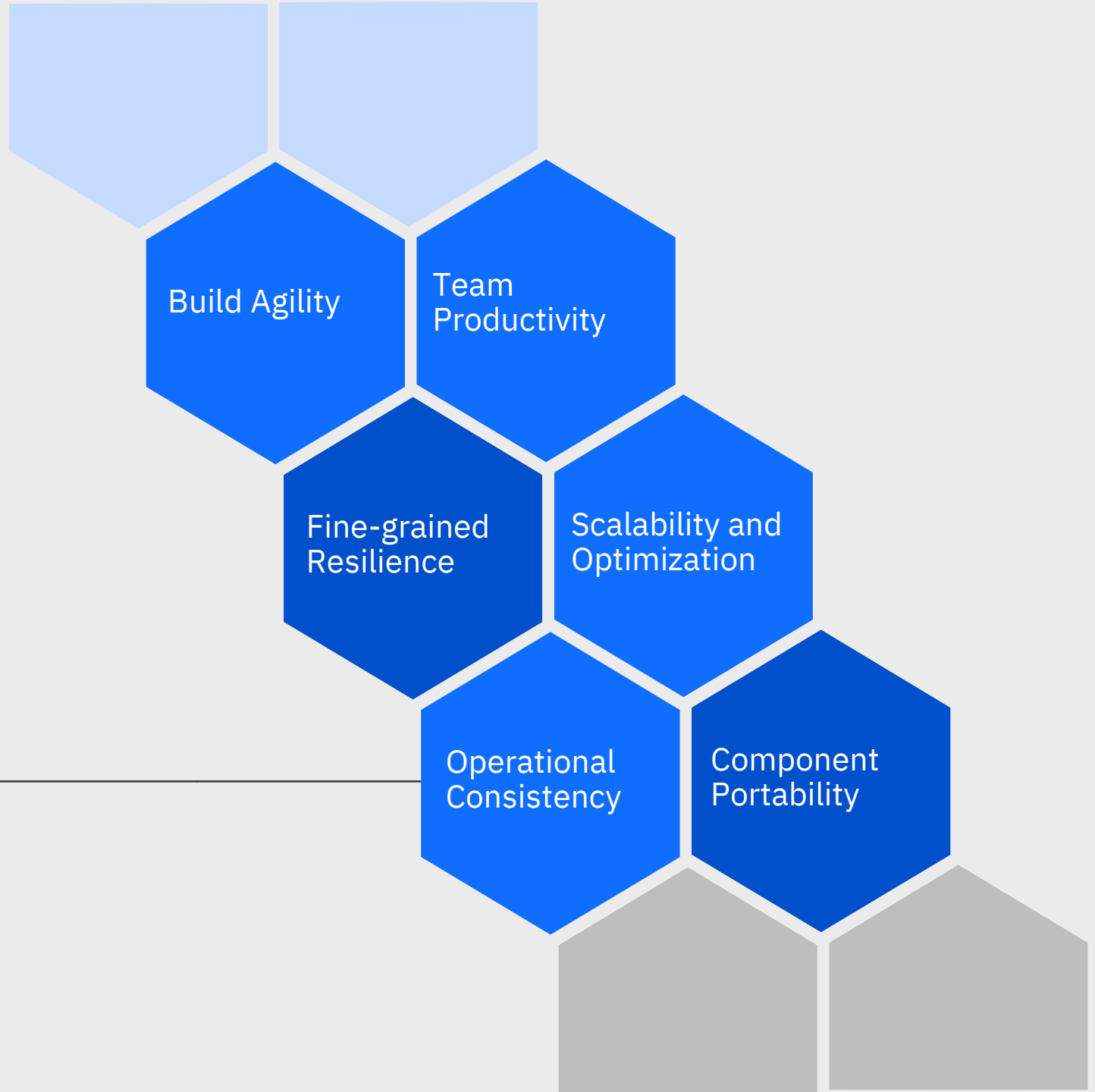


Cloud-native

Elastically scaled containers
Pooled shared underlying resources, but decoupled containers
Implicit HA/DR
Deploy by image combining artefacts and infrastructure
Administer image then redeploy, not hot fixing.



Benefits of a container based strategy

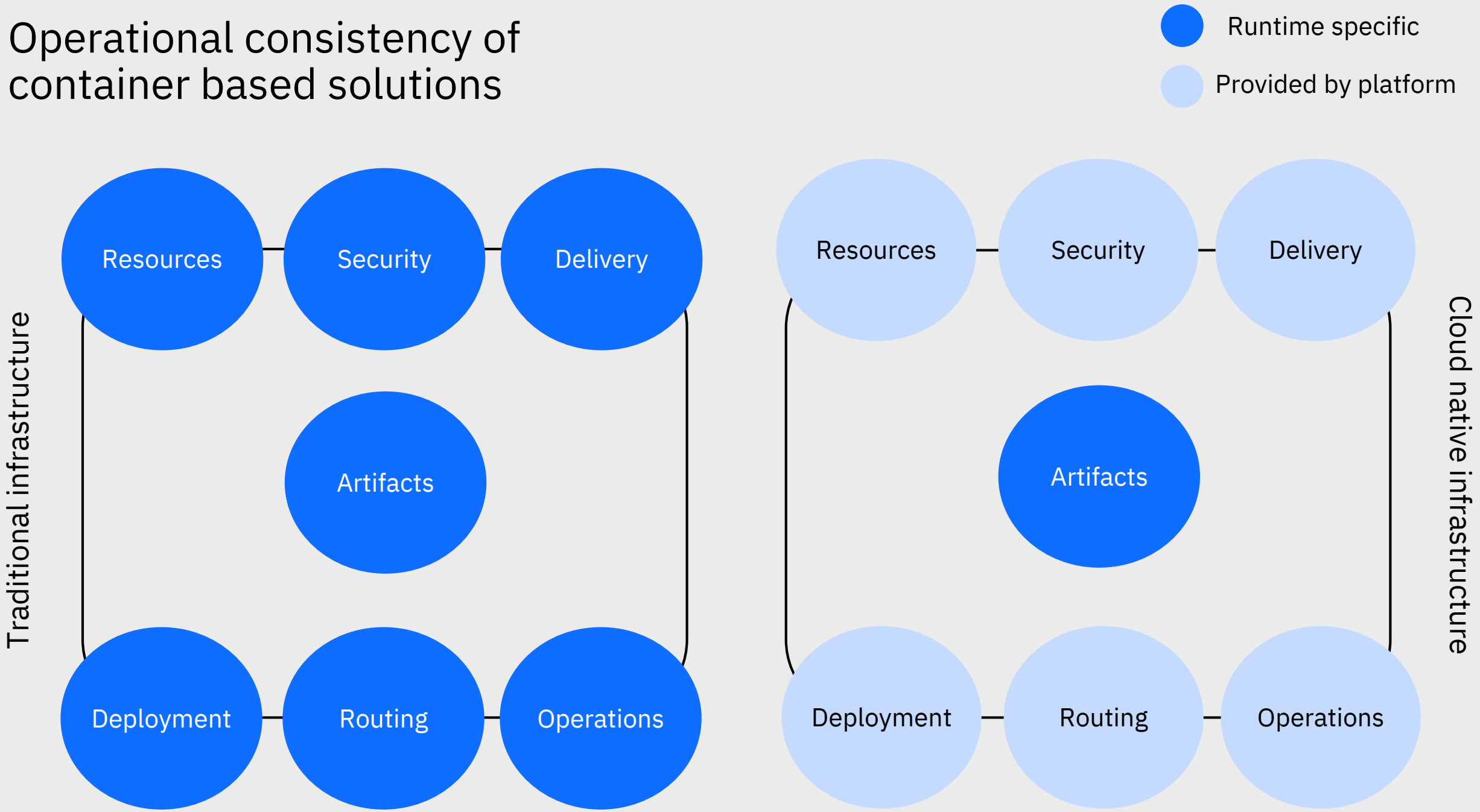


Operational Consistency

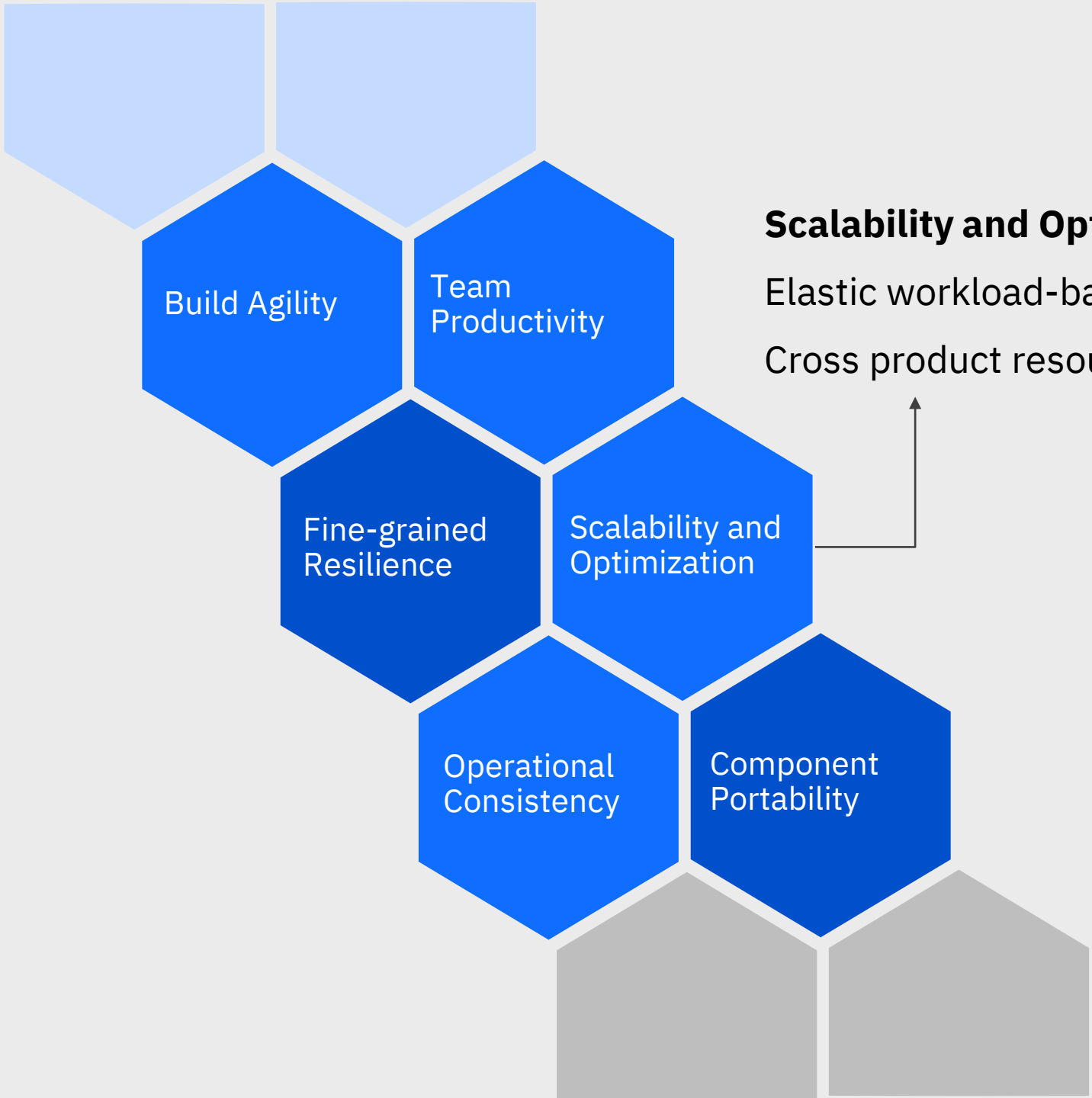
Single infrastructure skillset

Cross product consistency

Operational consistency of container based solutions



Benefits of a container based strategy

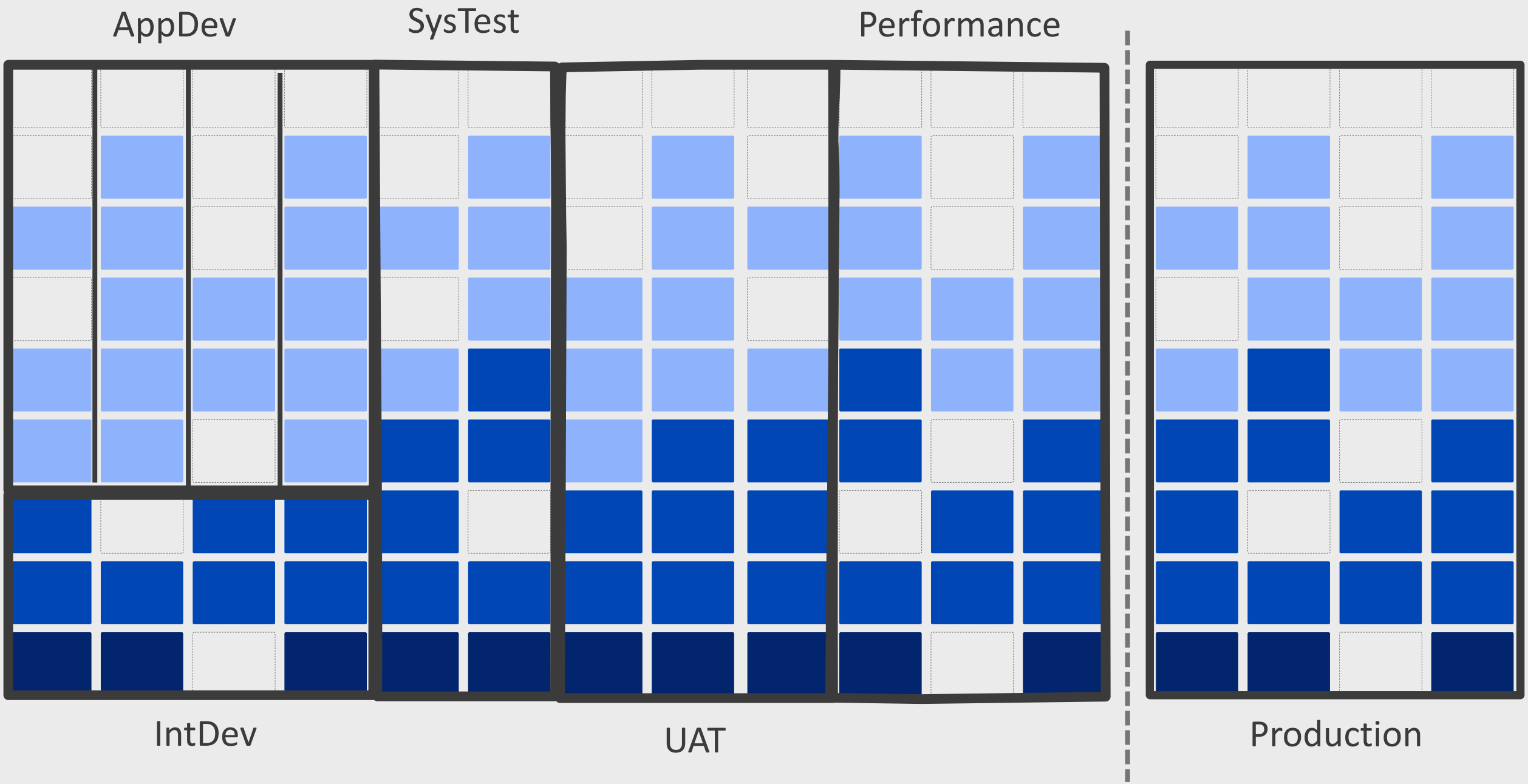


Scalability and Optimization

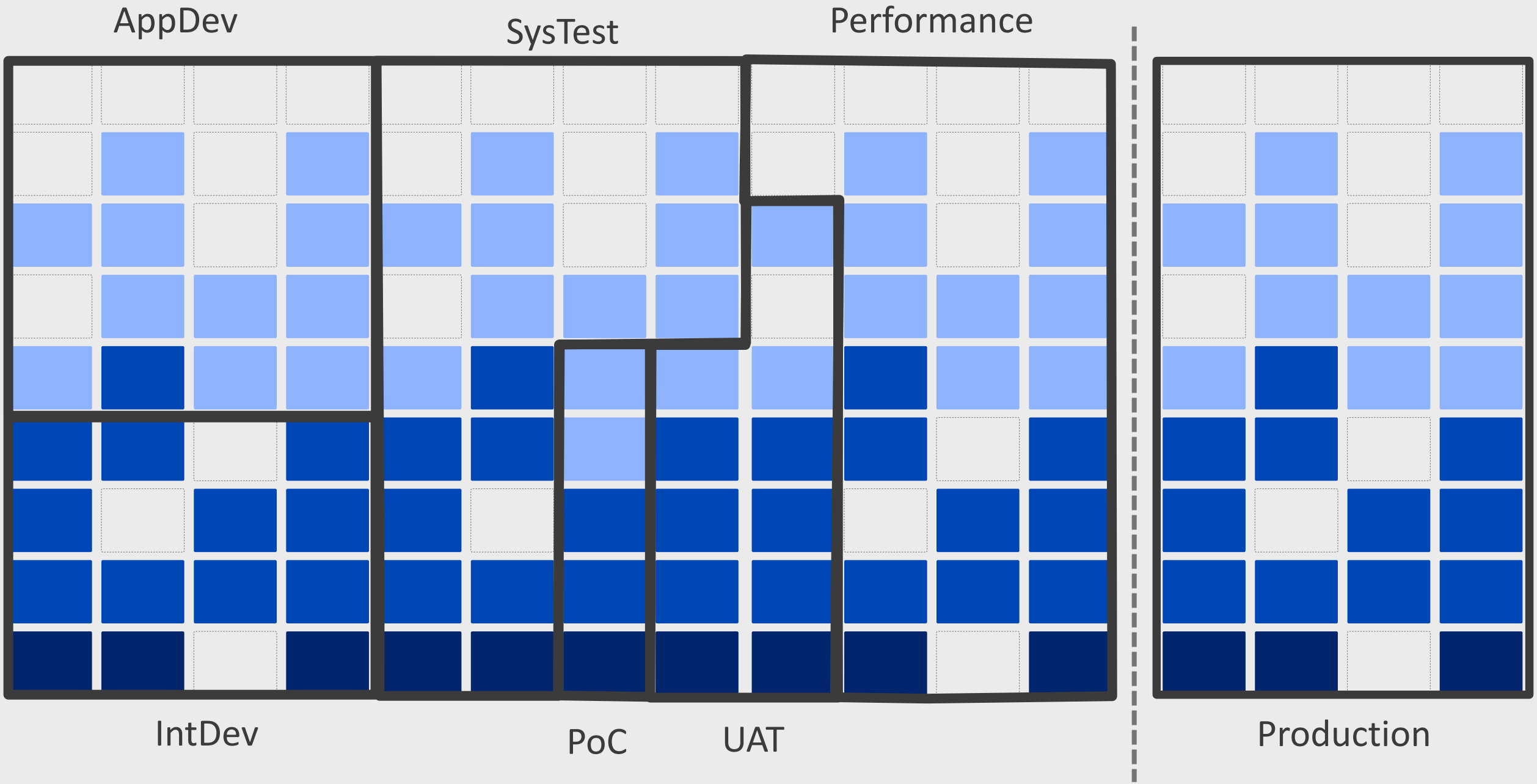
Elastic workload-based scaling

Cross product resource optimization

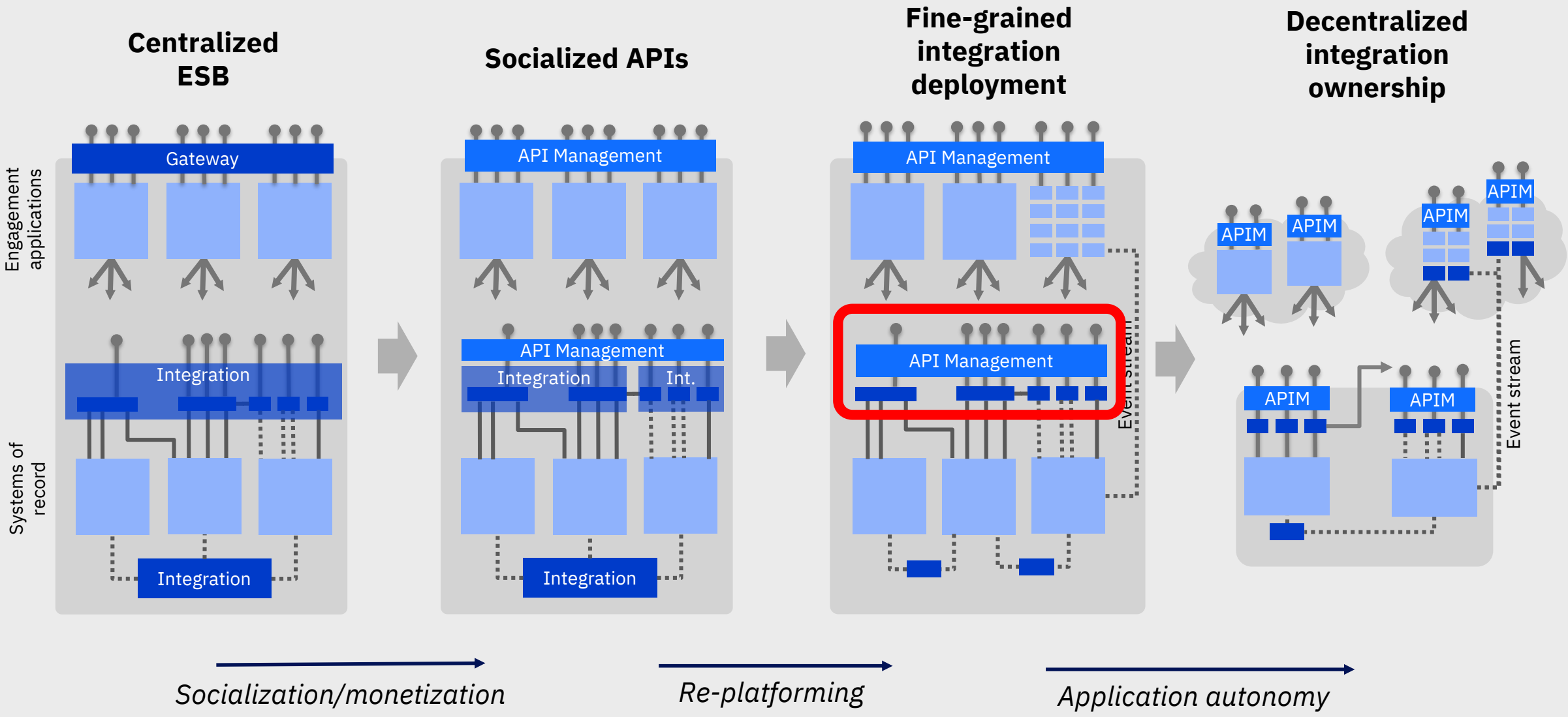
Elastically scaling containers across environments



Elastically scaling containers across environments



Evolution to **agile integration** – detail view

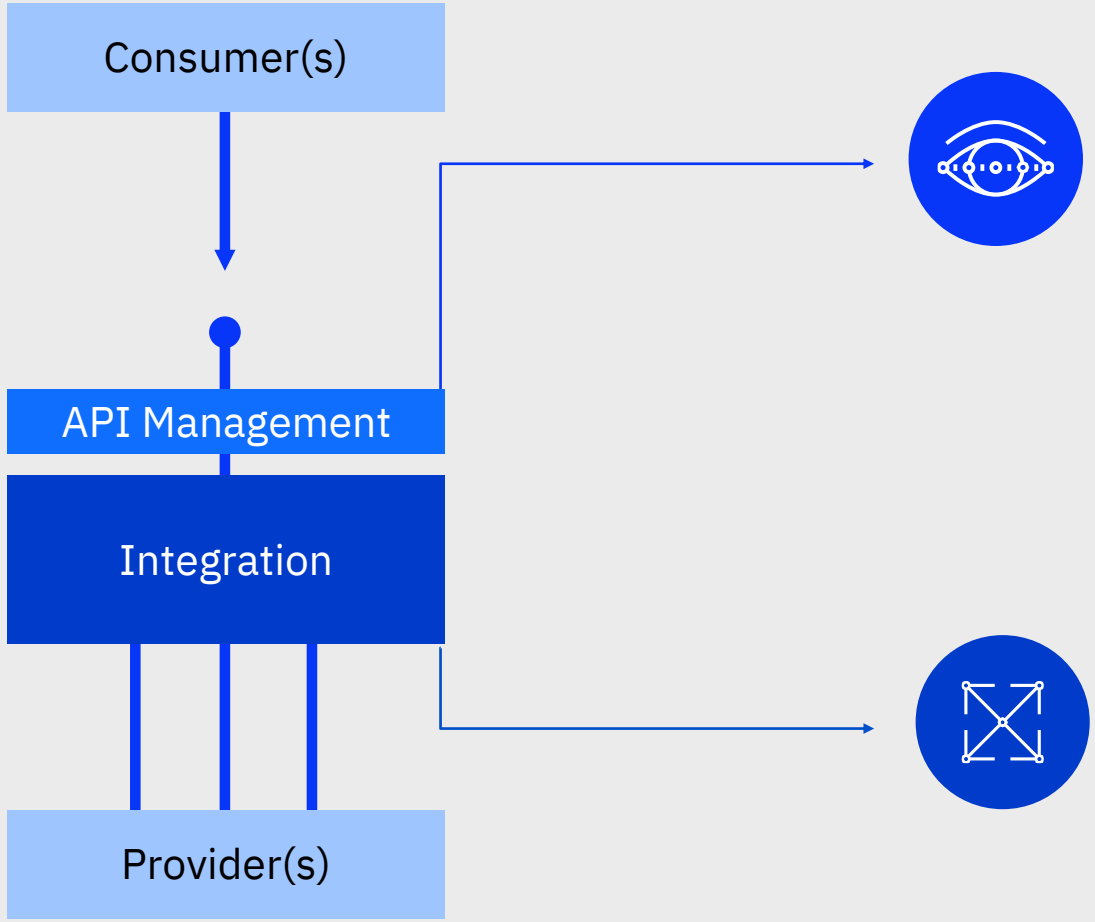


Webinars http://ibm.biz/agile-integration-webcasts

eBook http://ibm.biz/agile-integration-ebook

Section title

Differentiating exposure from implementation



Exposure
(consumer focused lifecycle)

Control point: Consistent provision of routing, versioning, traffic management, security, logging.

Socialization: Enables discovery, documentation, and self-subscription, analytics.

Implementation
(provider focused lifecycle)

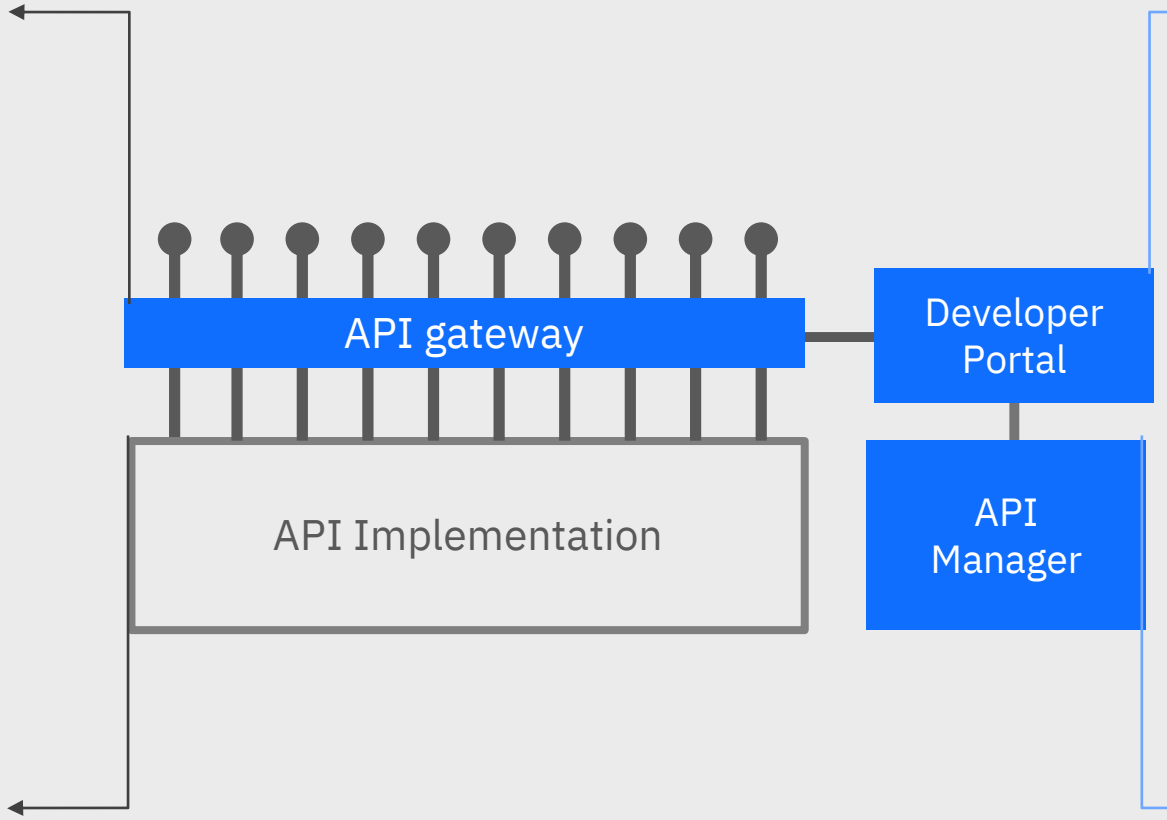
Composition: Implements the custom “integration logic”, including aggregation from multiple sources, and merging of data.

Adaptation: Understands the connectivity protocols and data formats, required to manage communication with specific provider systems.

API Management: More than just a gateway

API Gateway:
- Decoupling/routing
- Traffic management
- Security
- Translation

The API implementation should not be burdened with the complexities of API exposure beyond the microservices application boundary. Exposure should be delegated to a separate capability providing as a minimum, a gateway, a developer portal, and API management.

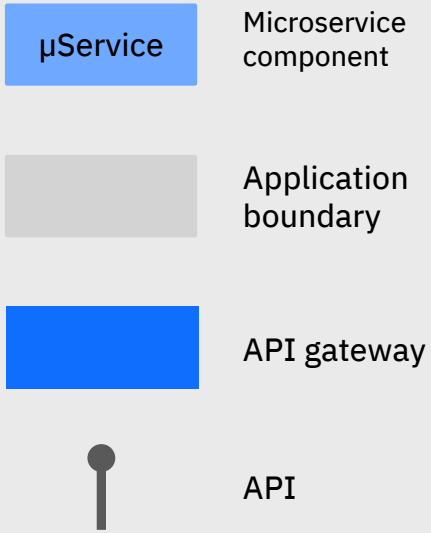


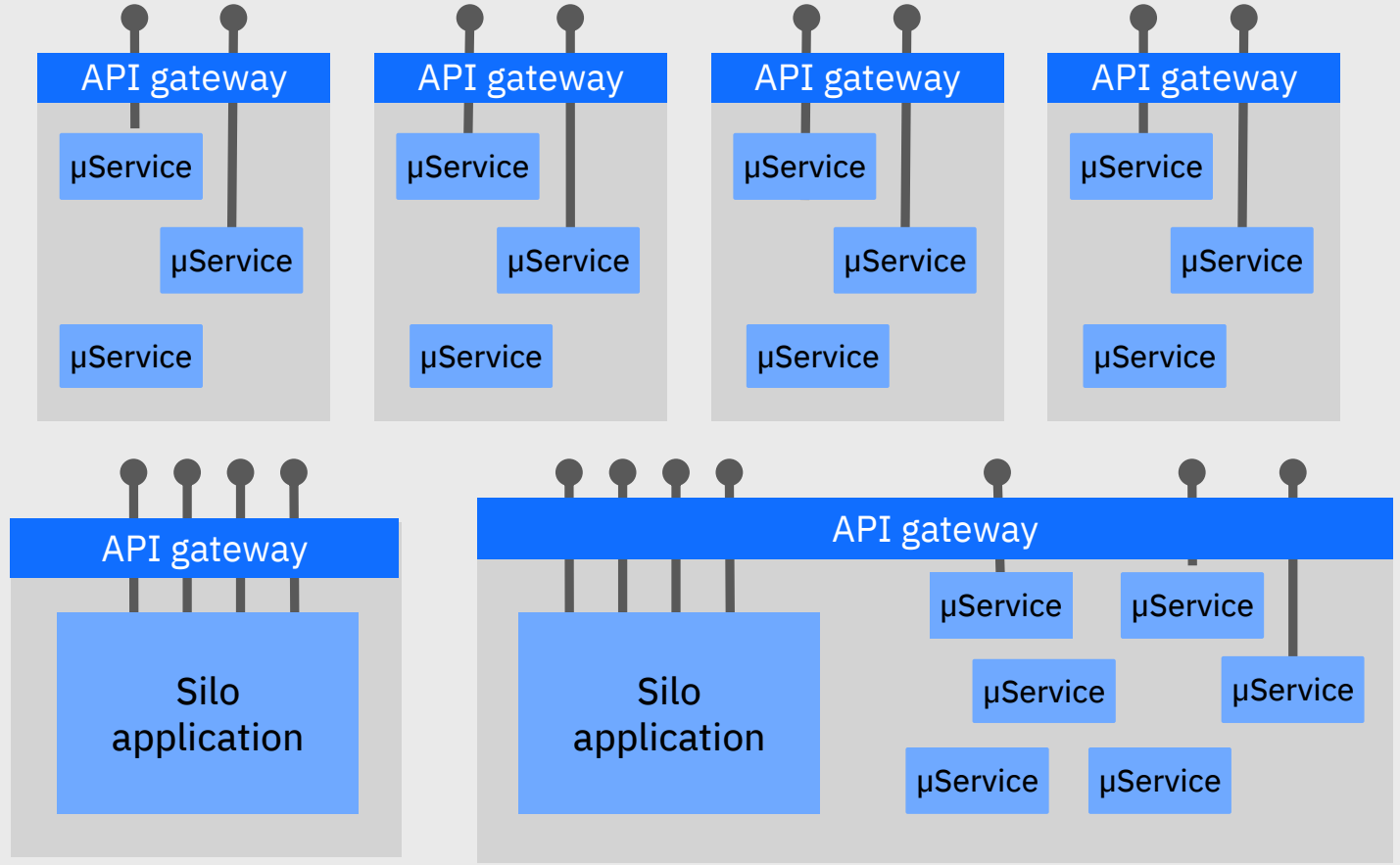
Developer portal:
- API discovery
- Self-service
  - Onboarding
  - API subscription
- Account usage analytics

API Manager:
- API/plan/product design
- Access management
- Policy administration
- API plan usage analytics

Boundaries make complex environments manageable

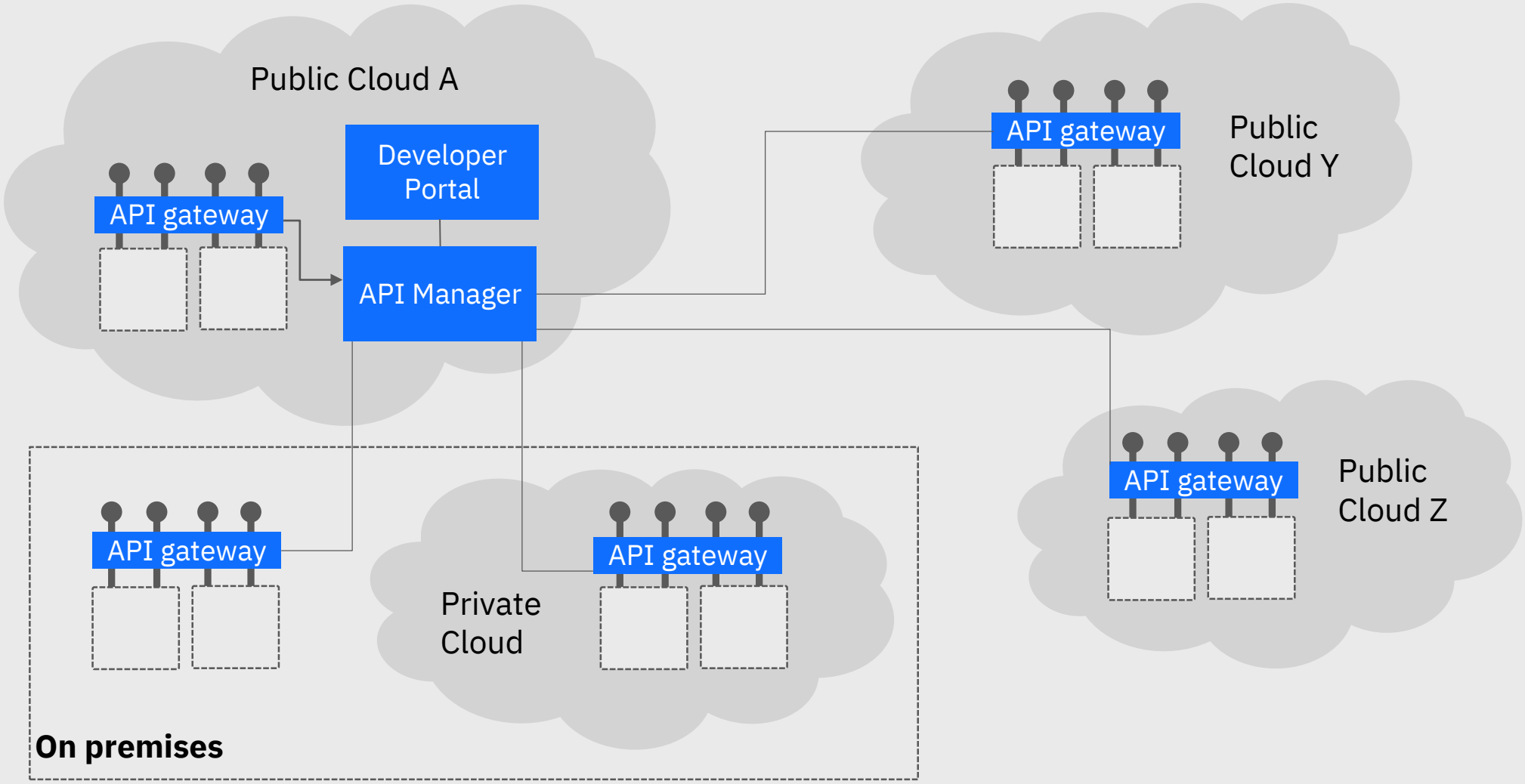
Managed API gateways define and enforce application boundaries



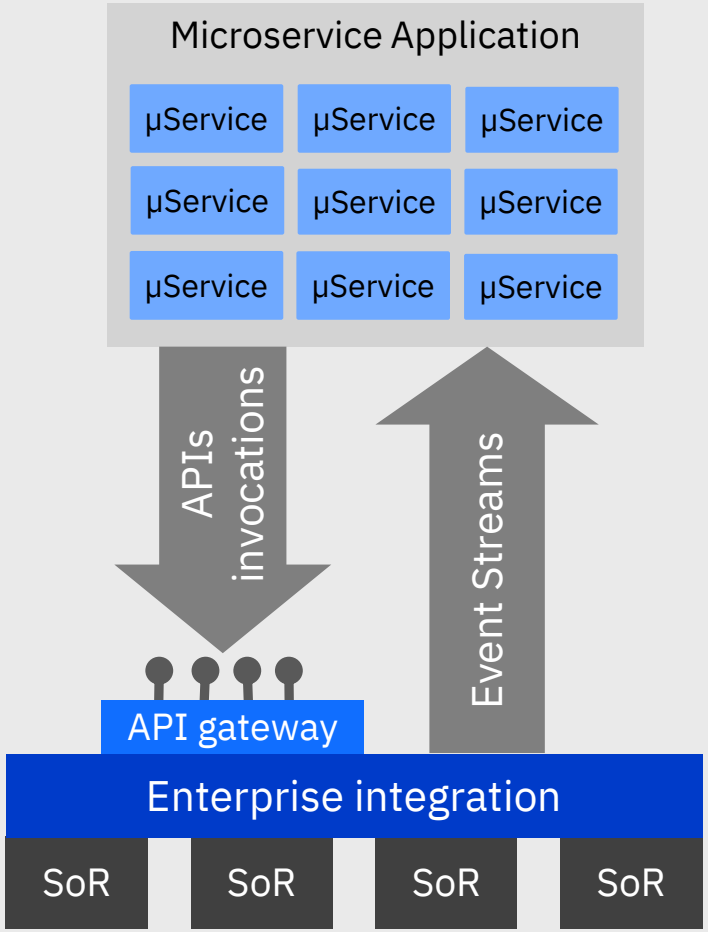


https://developer.ibm.com/apiconnect/2018/10/09/apis-microservices-defining-boundaries

Federated Management



Creating truly independent digital applications requires asynchronous communications as well as APIs



Truly independent, decoupled microservice components enable

Agility

Innovate rapidly without affecting other components

Scalability

Scale only what you need, and only when you need to

Resilience

Fail fast, return fast, without affecting other components

To provide those benefits they need to be independent of the systems of record

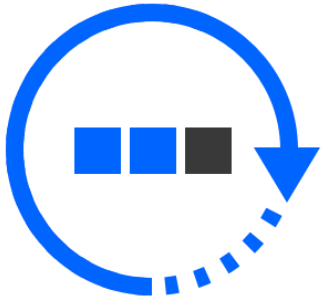
APIs

Are simplest to use, but create a real-time dependency

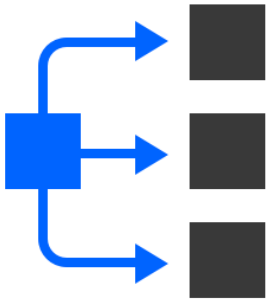
Event streams

Enable microservices to build decoupled views of the data

Events
(notifications)



Stream History

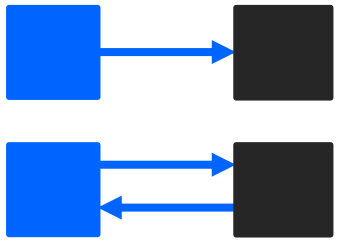


Scalable
Consumption

Immutable Data

Messaging
(commands)

Transient Data
Persistence



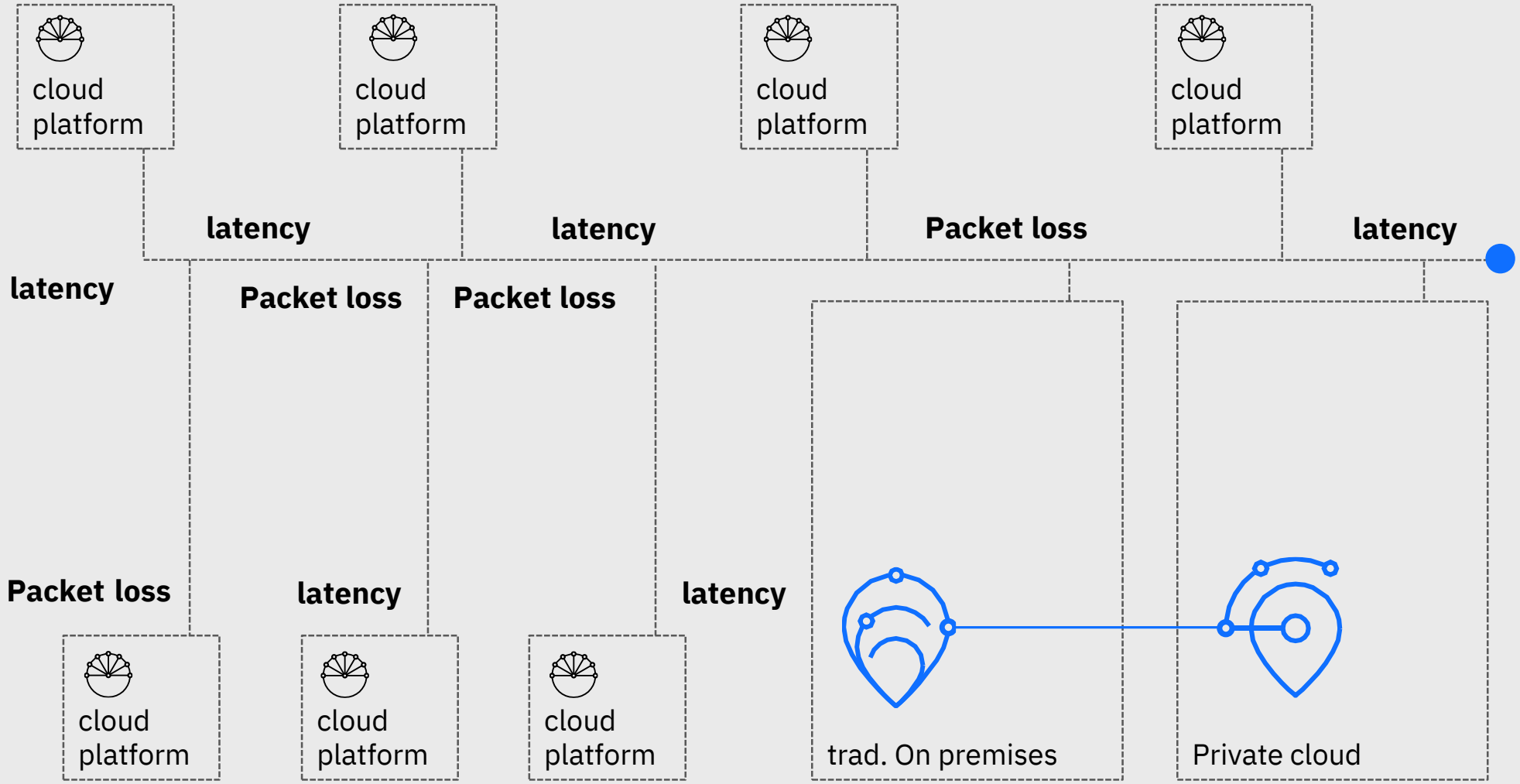
Source / Target
Request / Reply

Targeted
Reliable Delivery

https://developer.ibm.com/messaging/2018/05/18/comparing-messaging-event-streaming-use-cases

Modernizing batch jobs and file movement

As network distances increase, TCP suffers disproportionately with increases in packet loss and latency. For large file transfers this can result in order of magnitude increases in transfer time over FTP.



Moving to agile integration – a real world example

Moving to cloud is a progressive evolution of enterprise architecture, not a big bang Multiple aspects of integration architecture change along that journey

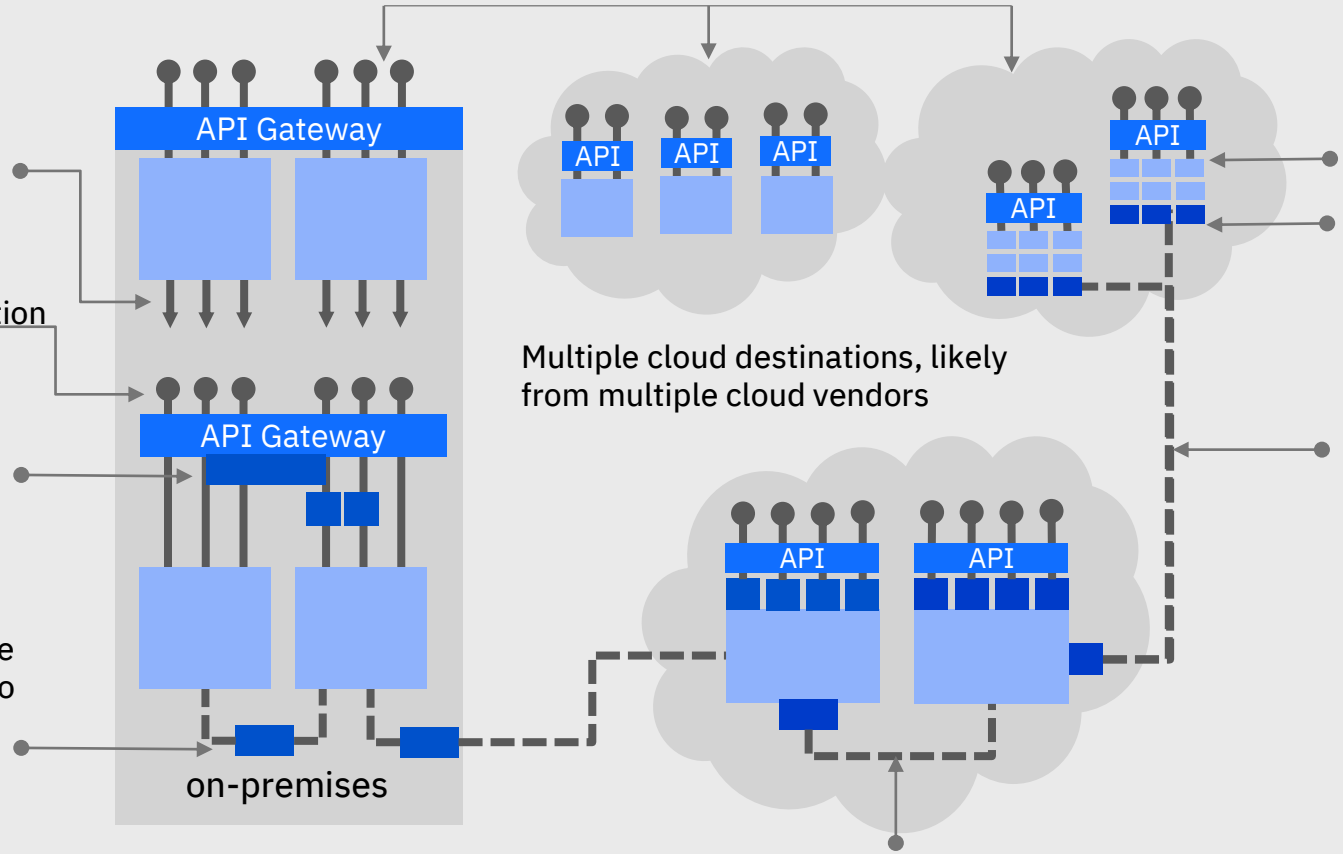
All applications, old and new, expose "managed" APIs such that they can be consumed by other applications

Engagement applications can consume any API exposed within the organization, or indeed beyond.

"Internal" APIs always require managed exposure, but use integration only where necessary

Existing ESB broken up into separately deployable integrations using containerization

Asynchronous "hub and spoke" style integrations such as "data sync" also split into separately deployable containers.



Multiple cloud destinations, likely from multiple cloud vendors

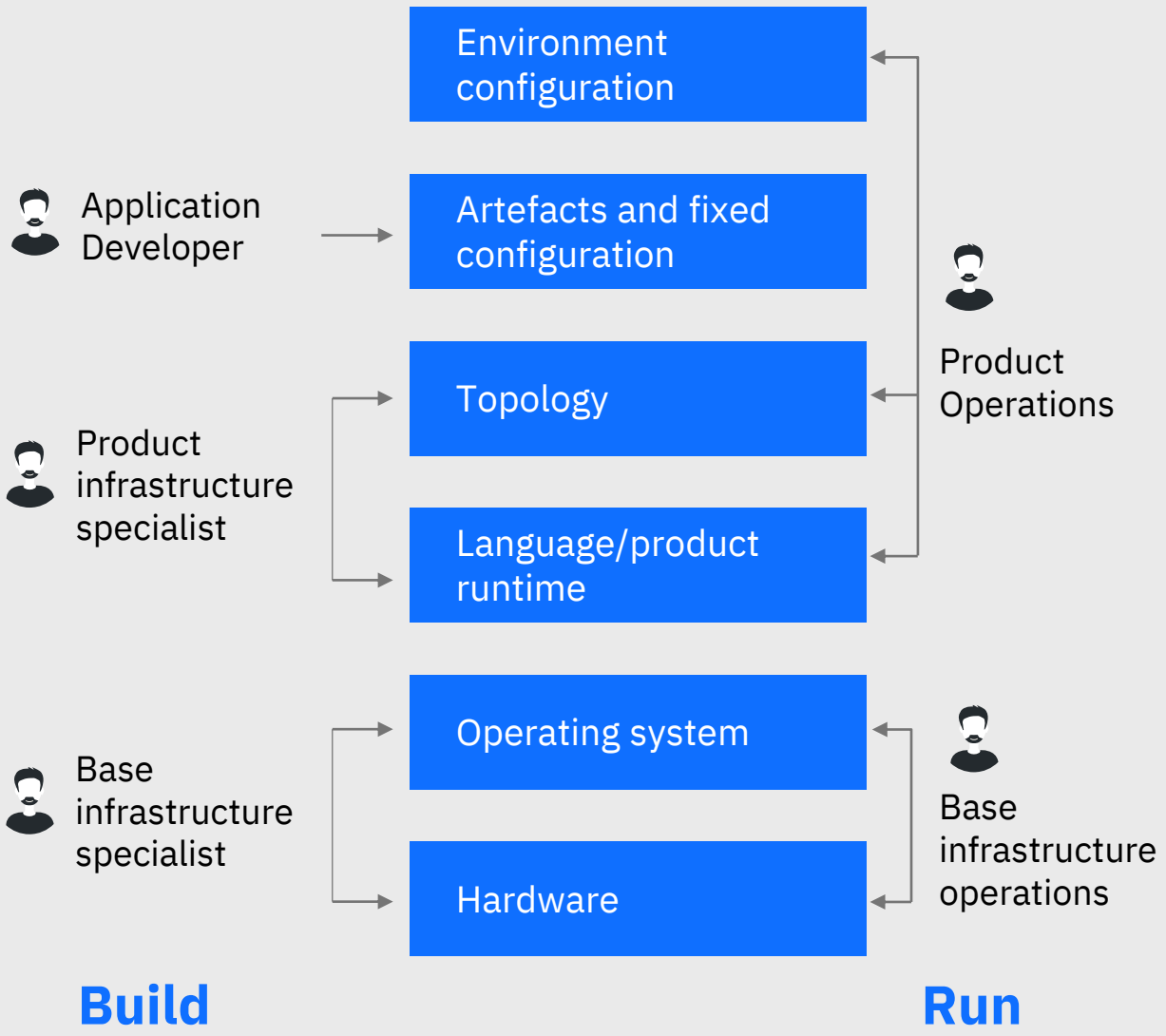
New applications created based on microservices architecture

Where a microservices performs an integration-like job, use a lightweight integration runtime

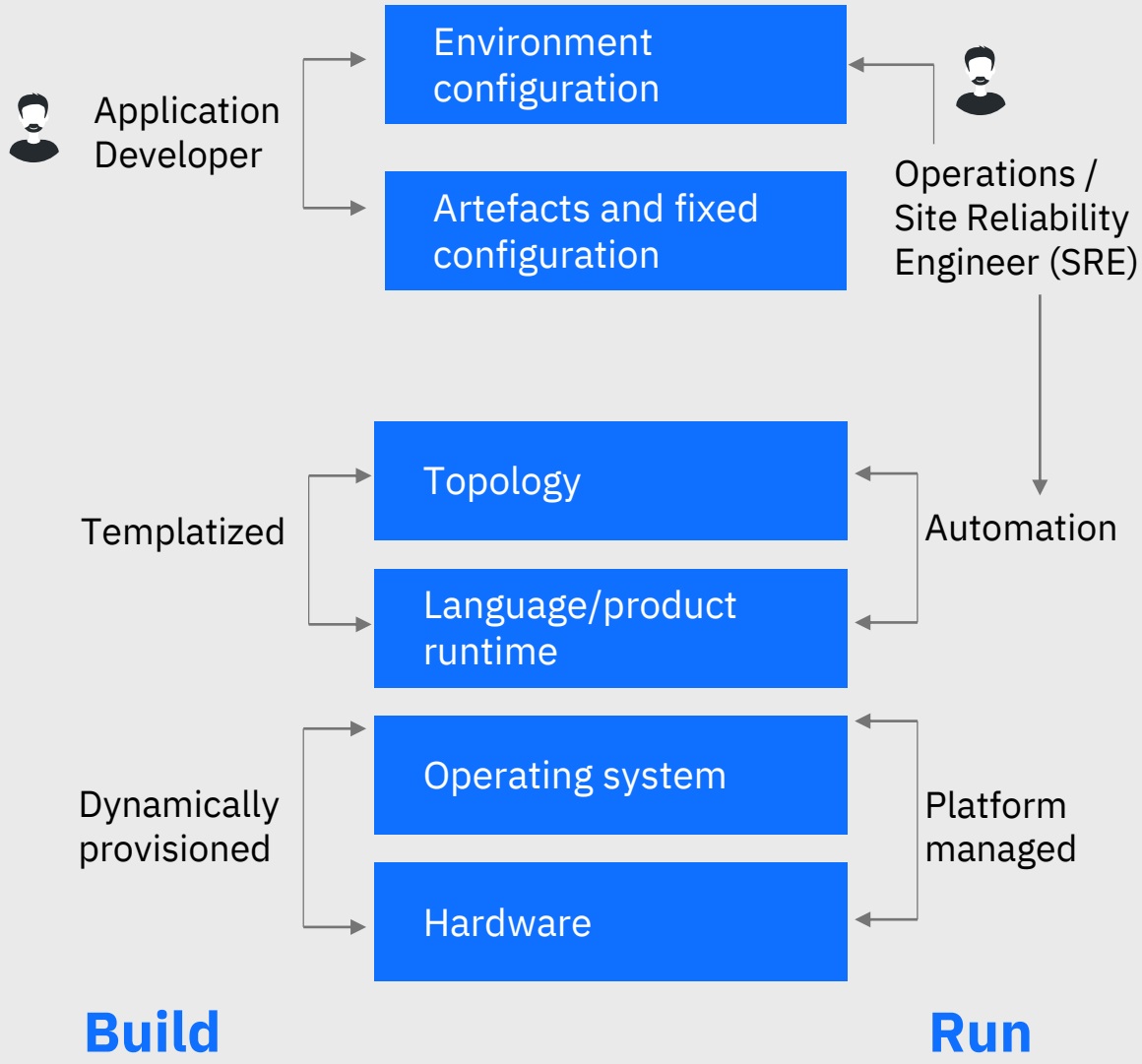
Event streams are made available such that microservice applications can build independent data stores rather than always having to make real-time calls over APIs

Applications migrated from on-premise to IaaS cloud, bring their integrations with them, potentially moving them under the ownership of the application teams

Traditional



Cloud native



Agile Integration

	Delivery focused architecture	**Decentralized ownership**	**Cloud native Infrastructure**
	Improve build independence and production velocity	Accelerate agility and innovation	Dynamic scalability and inherent resilience
Application Integration	Dependency free rapid integration delivery	Business autonomy for integration delivery	Scale and administer integrations with applications that live anywhere
API Lifecycle	Consumer centric exposure of business APIs	Self-administration of API exposure and subscription	Multi-platform cloud agnostic API management componentry
Messaging & Events	Independent application centric messaging	Self-provisioning of messaging and event capabilities	Cloud scale inherently resilient multi-platform messaging

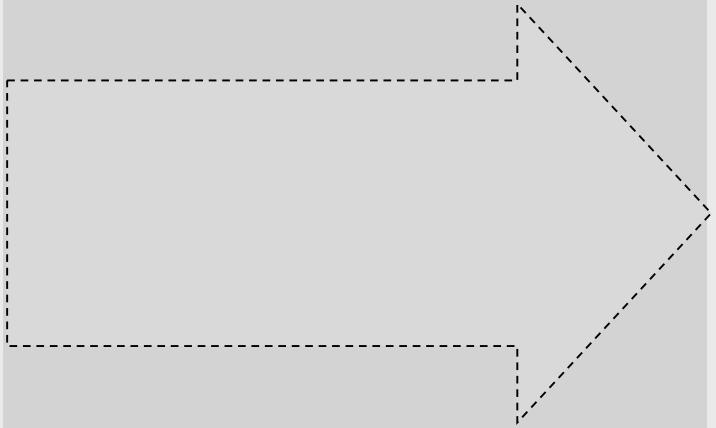
Traditional Integration

People & Process
Centralized
technology teams

Architecture
Centralized ESB

Technology
Nurtured
Environments

Business drivers
Innovation
and optimization



Outcomes
Development agility
Deployment agility
Operational agility

Agile Integration

People & Process

- Decentralized ownership
- Empowering teams
- Agile methods

Architecture:

- Fine-grained deployment
- API led
- Event-driven
- Microservices aligned

Technology:

- Cloud-native infrastructure
- Essential integration capabilities
- Unified security, governance and operations

Cloud Paks: Cloud-native middleware

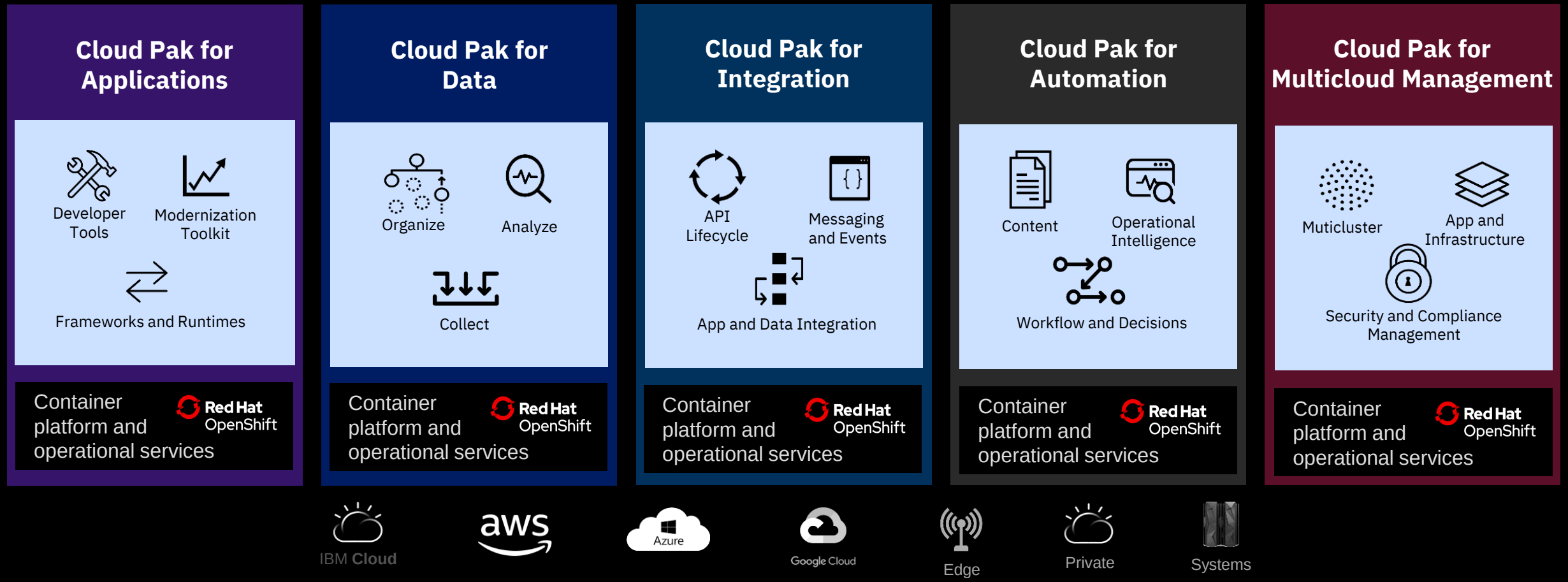
Enterprise-ready containerized software solutions

Complete yet simple
Application, data and AI services, fully modular and easy to consume

IBM certified
Full software stack support, and ongoing security, compliance and version compatibility

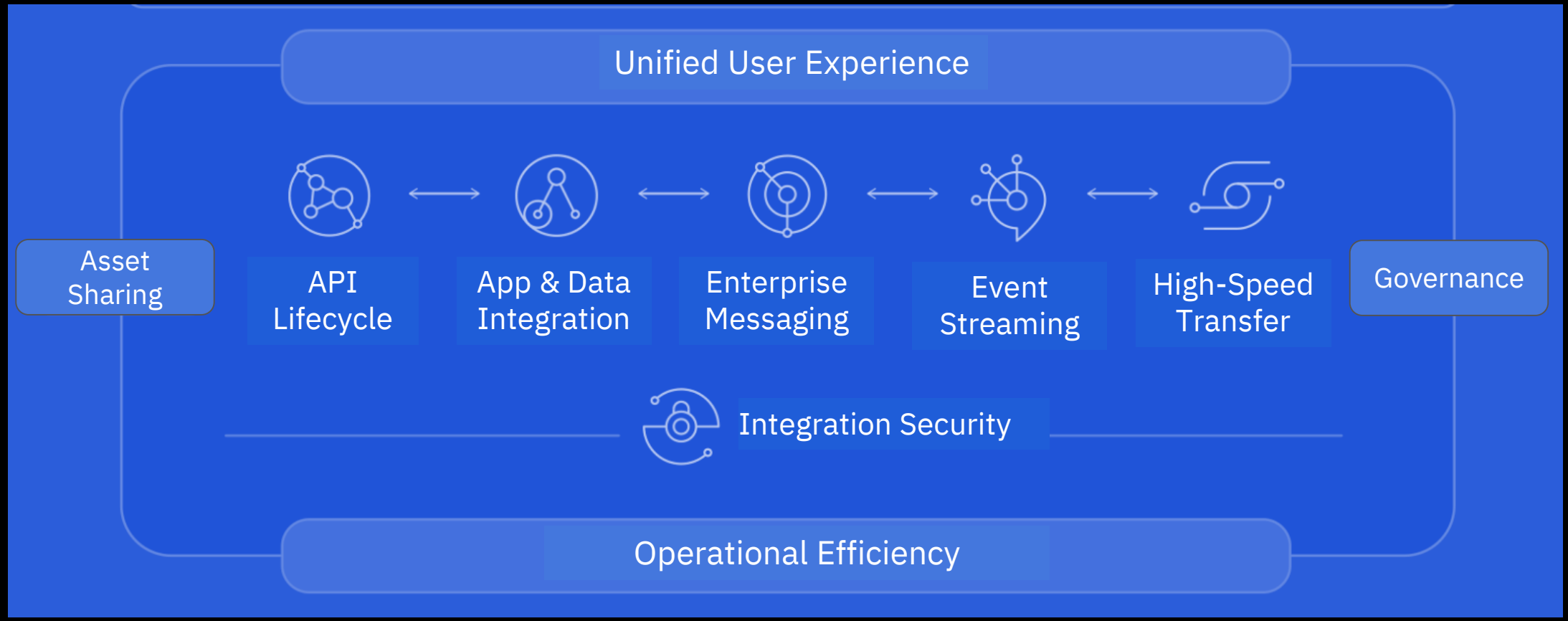
Run anywhere
On-premises, on private and public clouds, and in pre-integrated systems

Pre-integrated market-leading capabilities



Cloud Pak for Integration

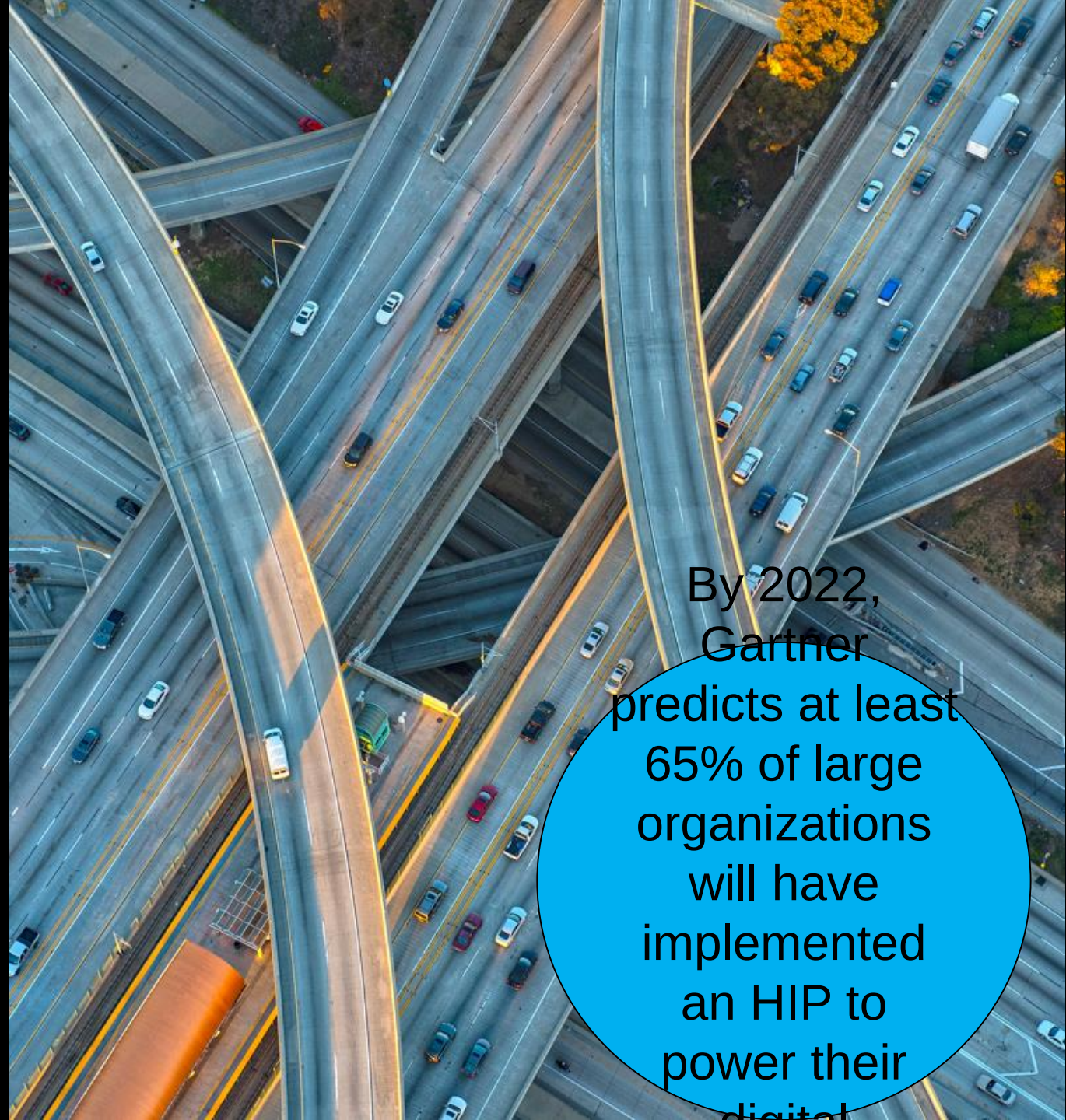
Enterprise-ready, containerized software solution including key integration capabilities to drive digital transformation



*Source: http://ibm.biz/Ovum-WP

How does Cloud Pak for Integration help

- Consistency across where apps and data live: on-prem, multiple clouds
- Streamlined development, architecture and operations
- Developer sharing and re-use of integration objects
- Unification of integration technologies simplifying solution building
- More participants in the integration development lifecycle



IBM Garage Services Agile Integration Workshop

- Modernize your integration architecture to address organization and governance requirements with blueprint for defining agile integration architecture.

Outcomes:

- Provide architecture guidance for integration modernization
- Identify requirements for new technologies
- Develop modern architecture and agile integration topologies
- Define organization and governance model
- Comprehensive report with outcomes
- Define next steps

Public material on integration modernization

Agile Integration

https://www.ibm.com/cloud/integration/agile-integration

eBooklet
http://ibm.biz/agile-integration-ebook

Webinar series
http://ibm.biz/agile-integration-webcasts

Other key links on agile integration
http://ibm.biz/agile-integration-links

Staying up to date:
https://developer.ibm.com/apiconnect/blog
https://developer.ibm.com/integration/blog
https://developer.ibm.com/messaging/blog

IBM Integration

https://developer.ibm.com/integration

Cloud Pak for Integration
https://www.ibm.com/cloud/cloud-pak-for-integration

Please submit your session feedback!

- Do it online at http://conferences.gse.org.uk/2019/feedback/JK
- https://www.ibm.com/developerworks/rfe/
- This session is JK