Accelerating Modernization with Agile Integration

Ricardo Ramos
Executive Integration Architect
Worldwide Cloud Integration SWAT Team

rramos@pe.ibm.com

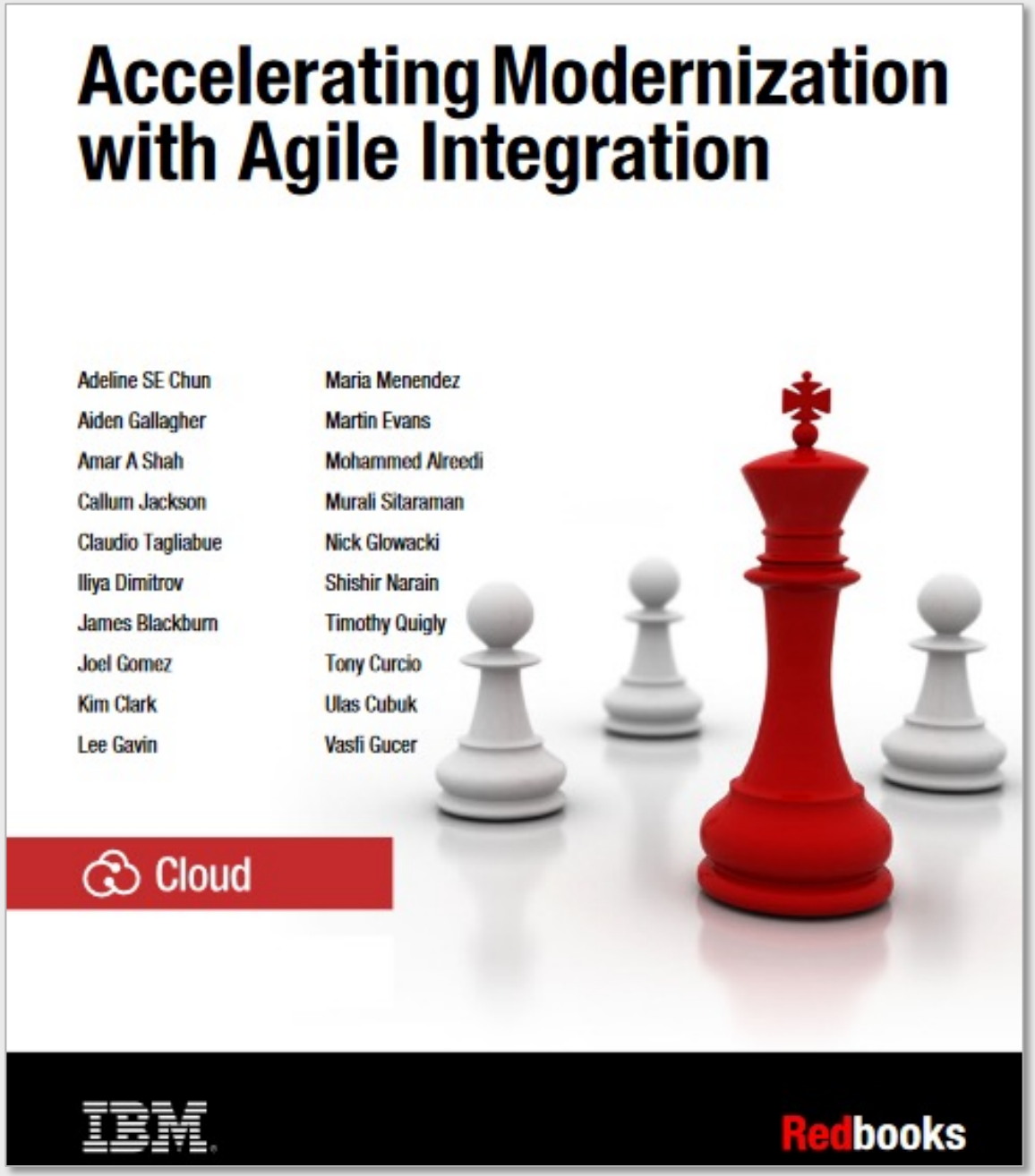
IBM Cloud

14th May (this presentation!): Accelerating Modernization with Agile Integration

Agile integration is an evolving approach to integration that ensures it remains the enabler of innovation, rather than a bottleneck. What does it mean to embrace agility, containerization, and cloud native principles in the integration space? How do we build more fine-grained, scalable, and portable integration components. What would rapid self-provisioning of integration capabilities like API management, event streams, messaging look like? How do you retain control in a decentralized world? How would we organise teams and roles differently in order to make all this these changes sustainable? Based on the recent IBM Redbook on agile integration http://ibm.biz/agile-integration-redbook

28th May: The Convergence of Integration and Application Development

Innovative applications today are rarely self contained. They are fundamentally dependent on the ability to bring disparate data together in new and unique ways. This means integration is at the core of all new applications. In the past, the creation of integrations and applications have been different disciplines. Nowadays, application developers regularly perform integration when defining and exposing their own APIs and events. Integration capabilities are now simply part of application developer's toolkit. We'll discuss how this is resulting in a new generation of powerful integration-enabled applications.



Explores the merits of agile integration, a container-based, decentralized and microservice aligned approach for integration solutions that meets the demands of agility, scalability and resilience required by digital transformation.

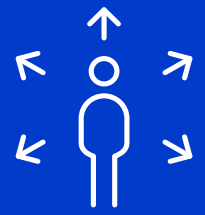
Practical examples based on
IBM Cloud Pak for Integration

Table of contents

http://ibm.biz/agile-integration-redbook

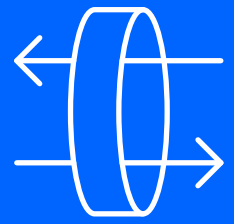
Agile Integration....

...to achieve development, deployment, and operational agility



People & Process

- Decentralized ownership
- Empowering teams
- Agile methods



Architecture

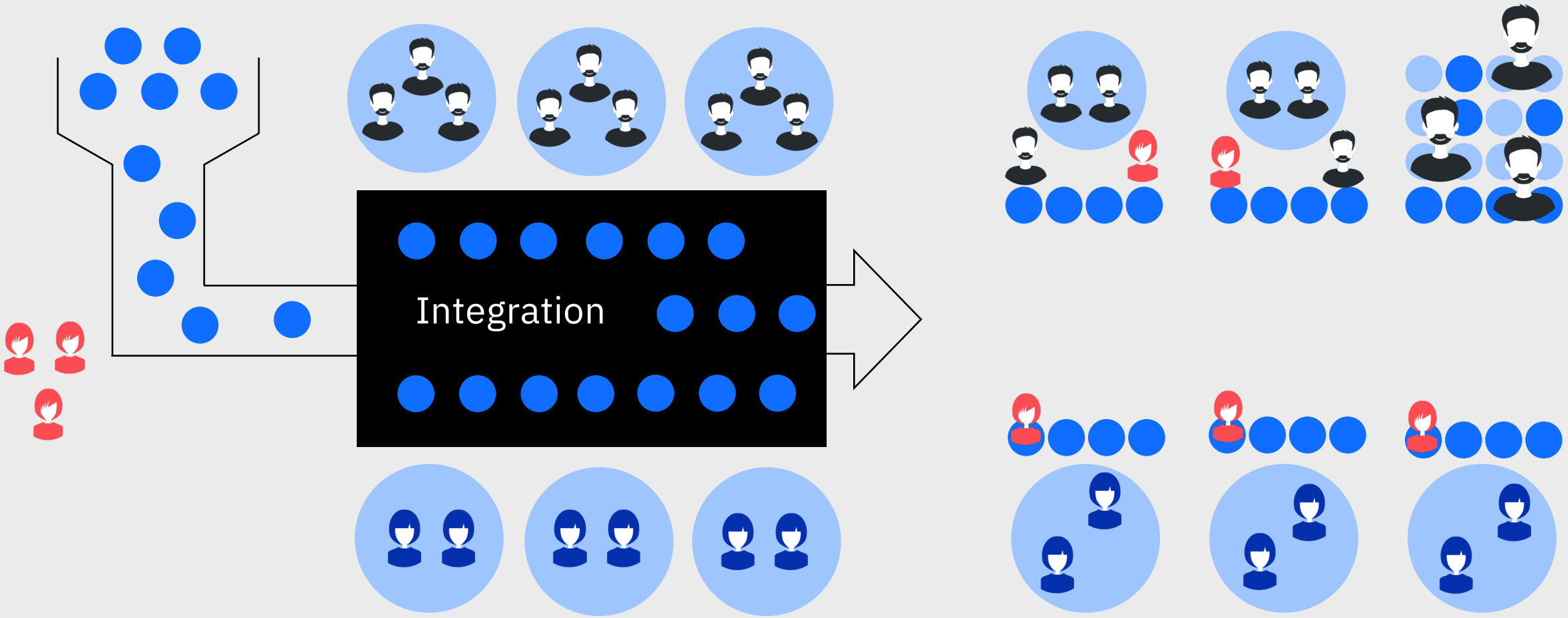
- Fine-grained deployment
- API led
- Event-driven
- Microservices aligned
- Highly scalable

Technology

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

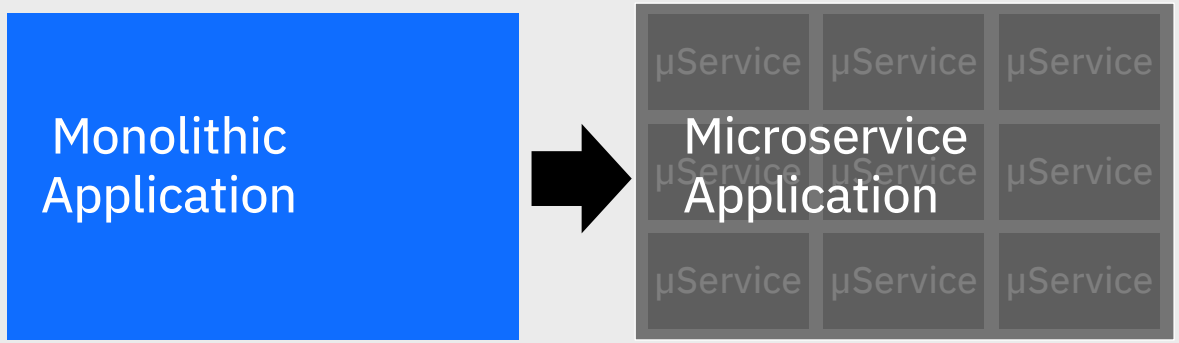
People & Process

Decentralized Ownership



https://www.ibm.com/blogs/cloud-computing/2017/04/11/decentralized-integration-innovation

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

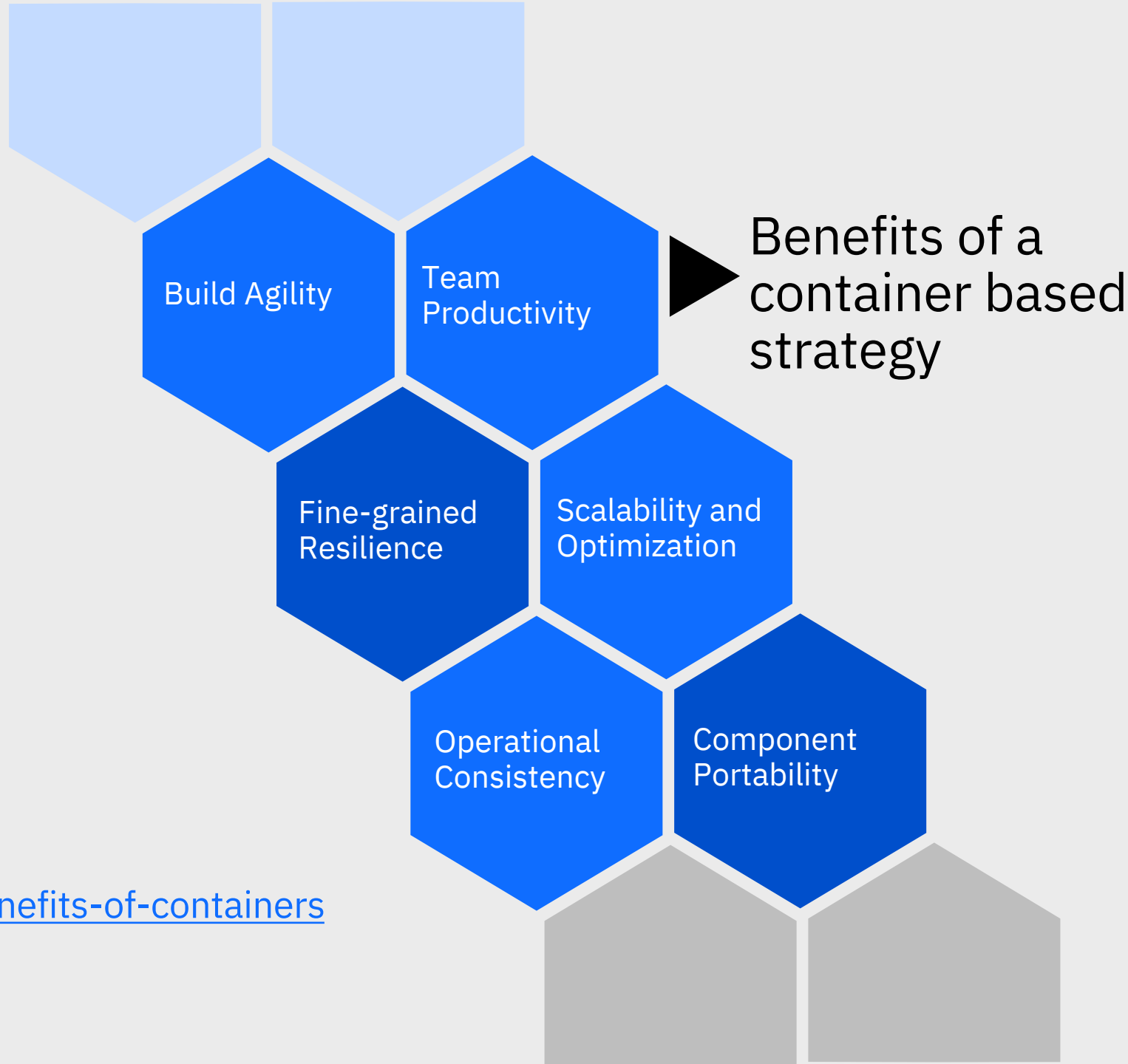
However, microservices is just one of architecture and design influences changing the way we think about building components.
API led, microservices, cloud-native, event driven...the list continues

Technology

Move to cloud is much more than re-platforming.

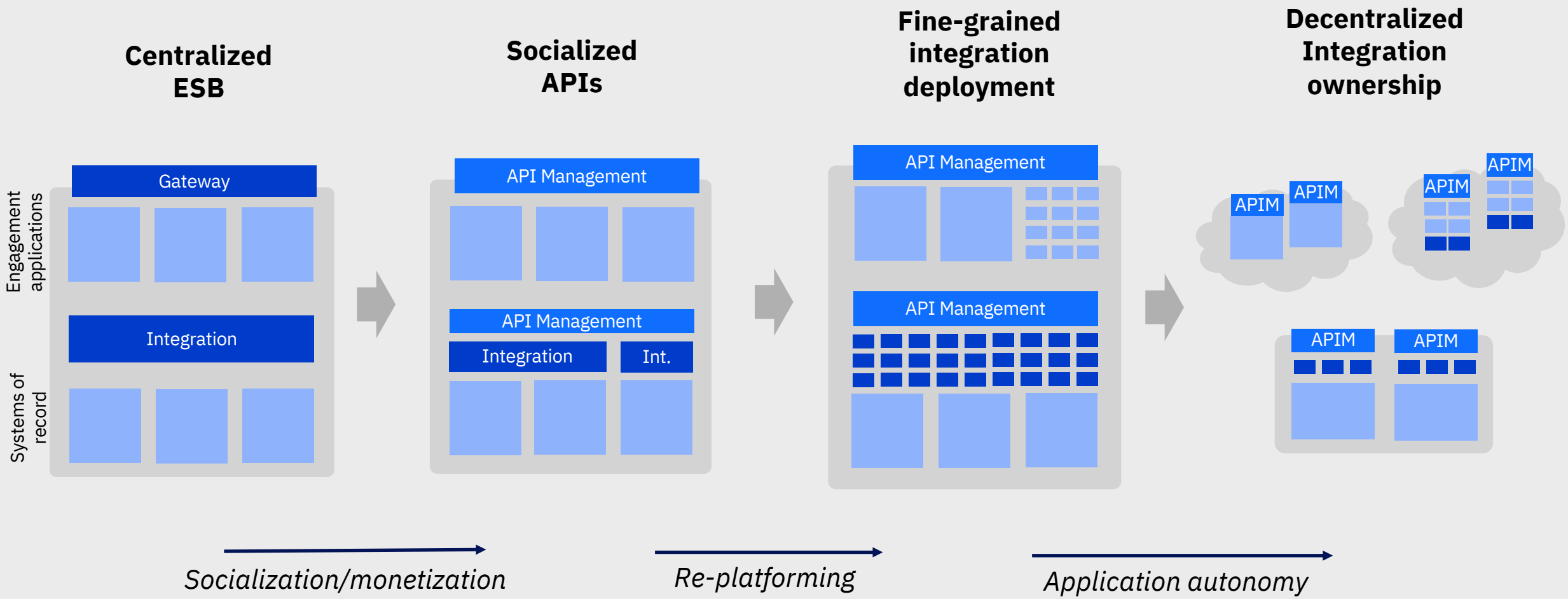
Containers, used in a cloud-native style are part of an evolving story.

Lift and shift will not bring same benefits



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

Evolution to **agile integration** – high level view



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

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

IBM Redbook http://ibm.biz/agile-integration-redbook

Benefits of a container based strategy

Fine-grained Resilience

Disposable components

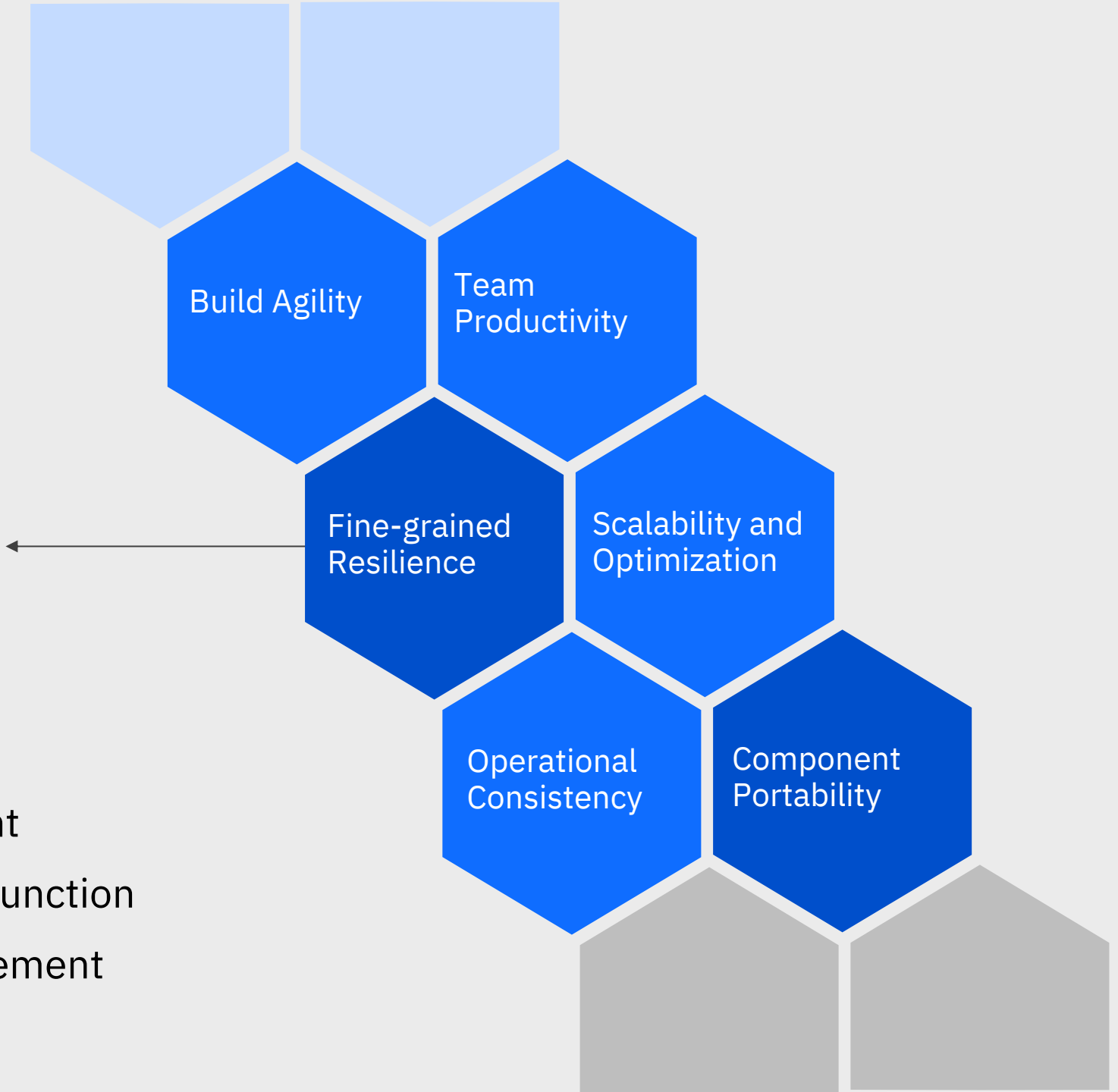
Rapid re-start

Fine-grained roll-out

Independent deployment

Discrete topologies per function

Implicit HA via re-instatement

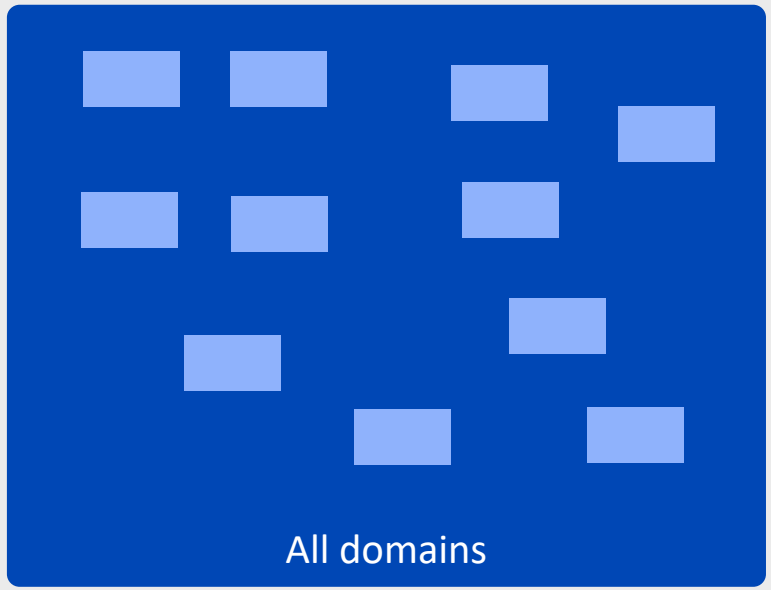


Deployment granularity

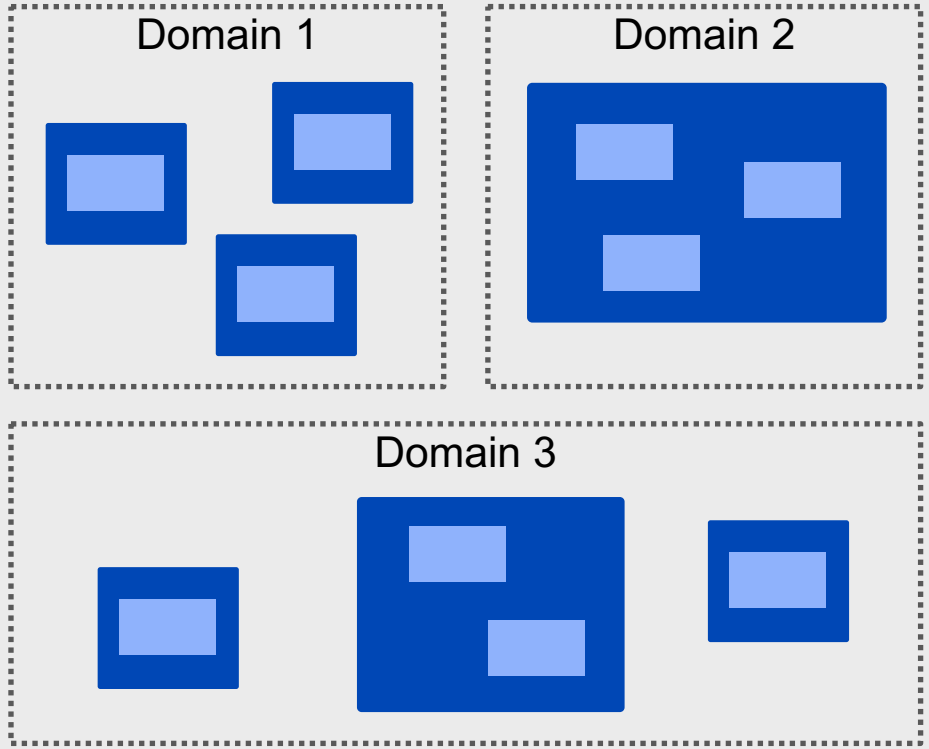
integration artefact

integration runtime

before



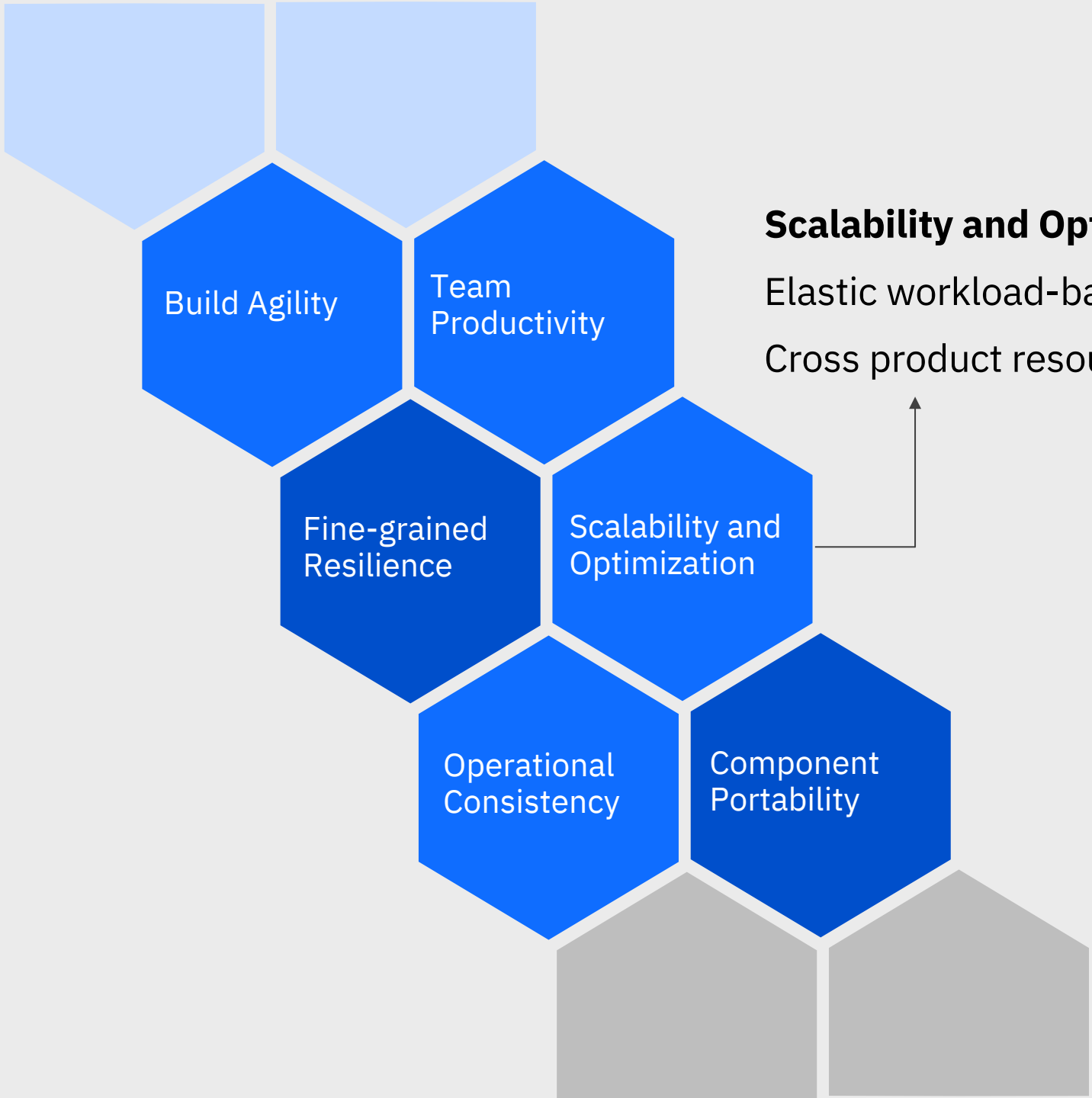
after



First split by business domains and functional areas to ensure high-level autonomy.

Next, consider non-functionals such as a) which need a separate pipeline (for **agility**), b) which need independent **scalability**, c) which have unique **resilience** requirements

Benefits of a container based strategy

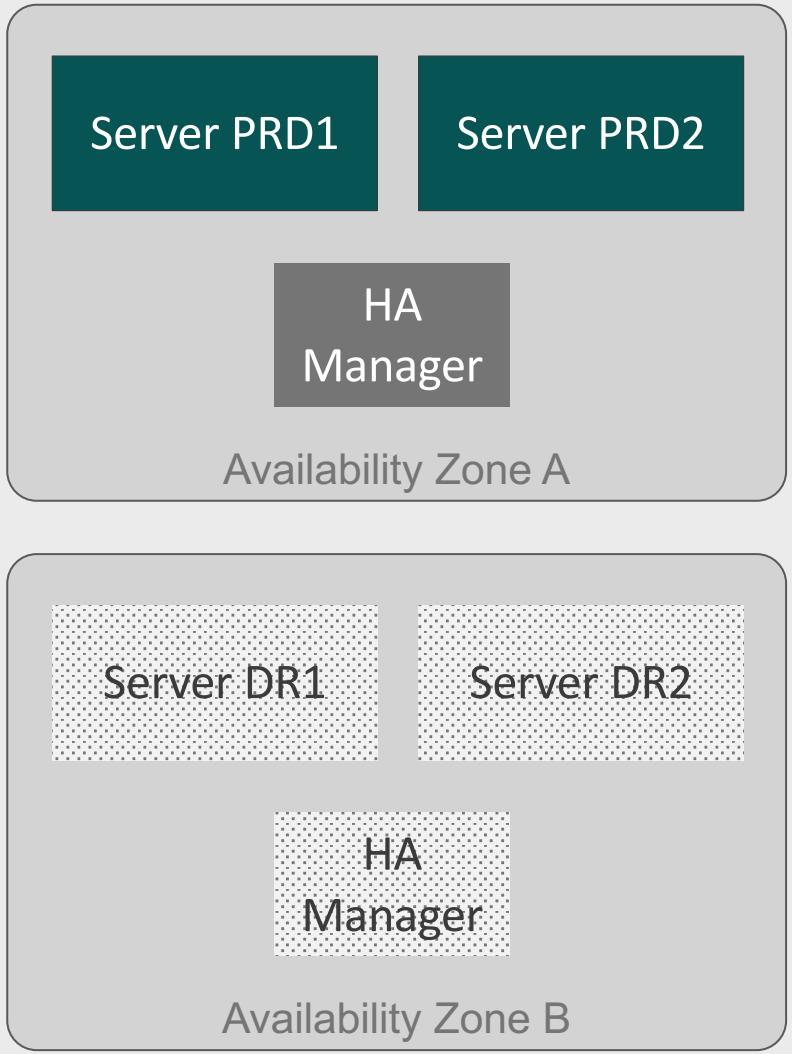


Scalability and Optimization

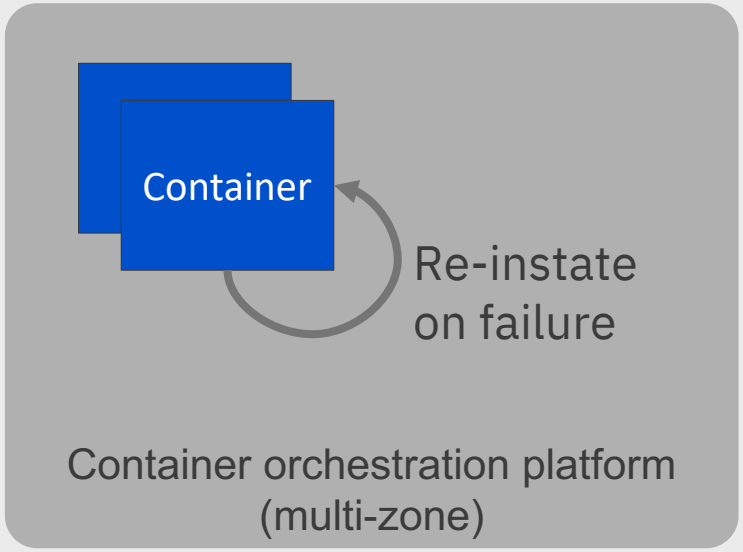
Elastic workload-based scaling

Cross product resource optimization

What's the container equivalent of the HA/DR topology you have today?



Traditional
(explicit configuration)

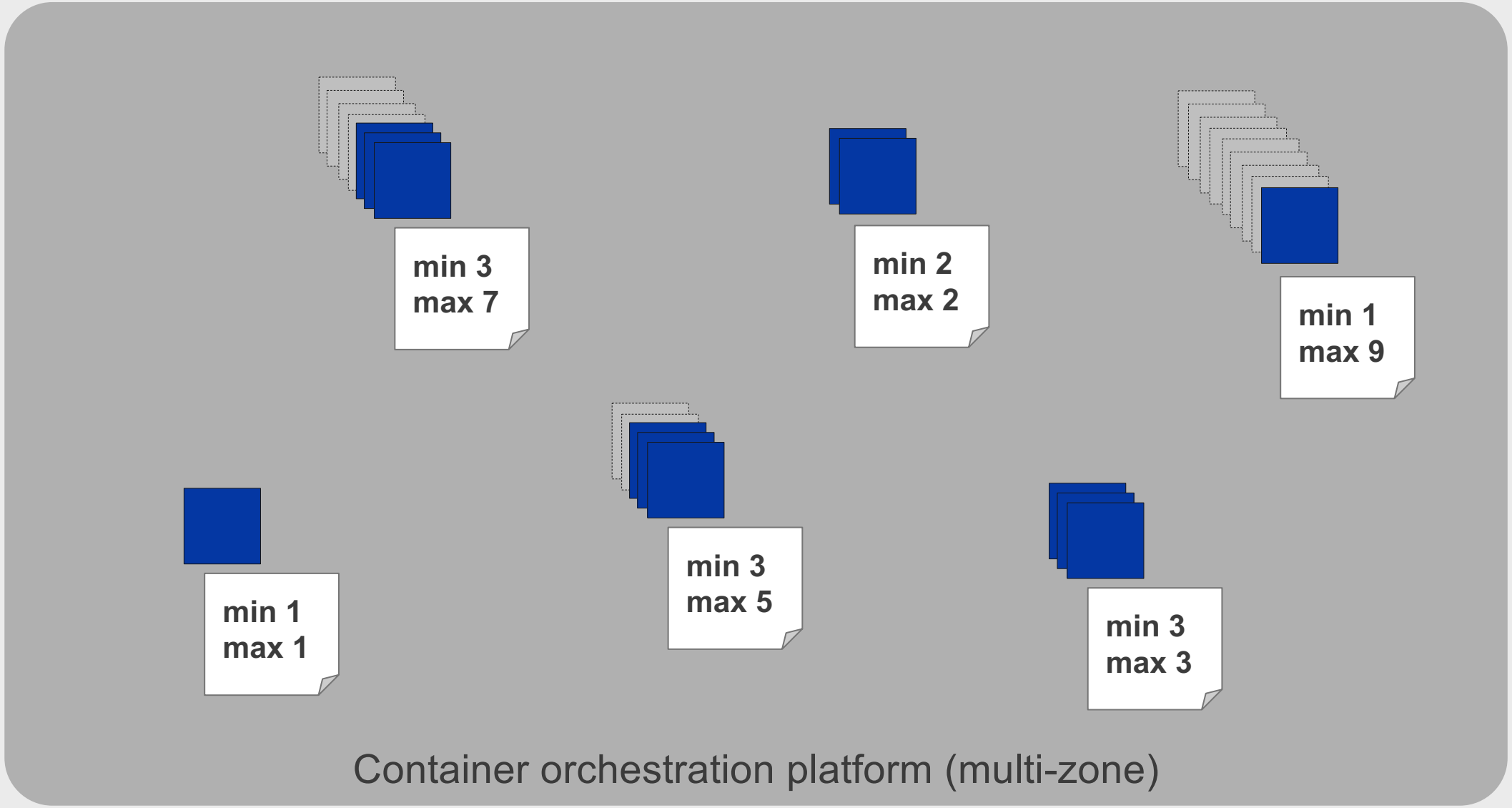


Infrastructure as code

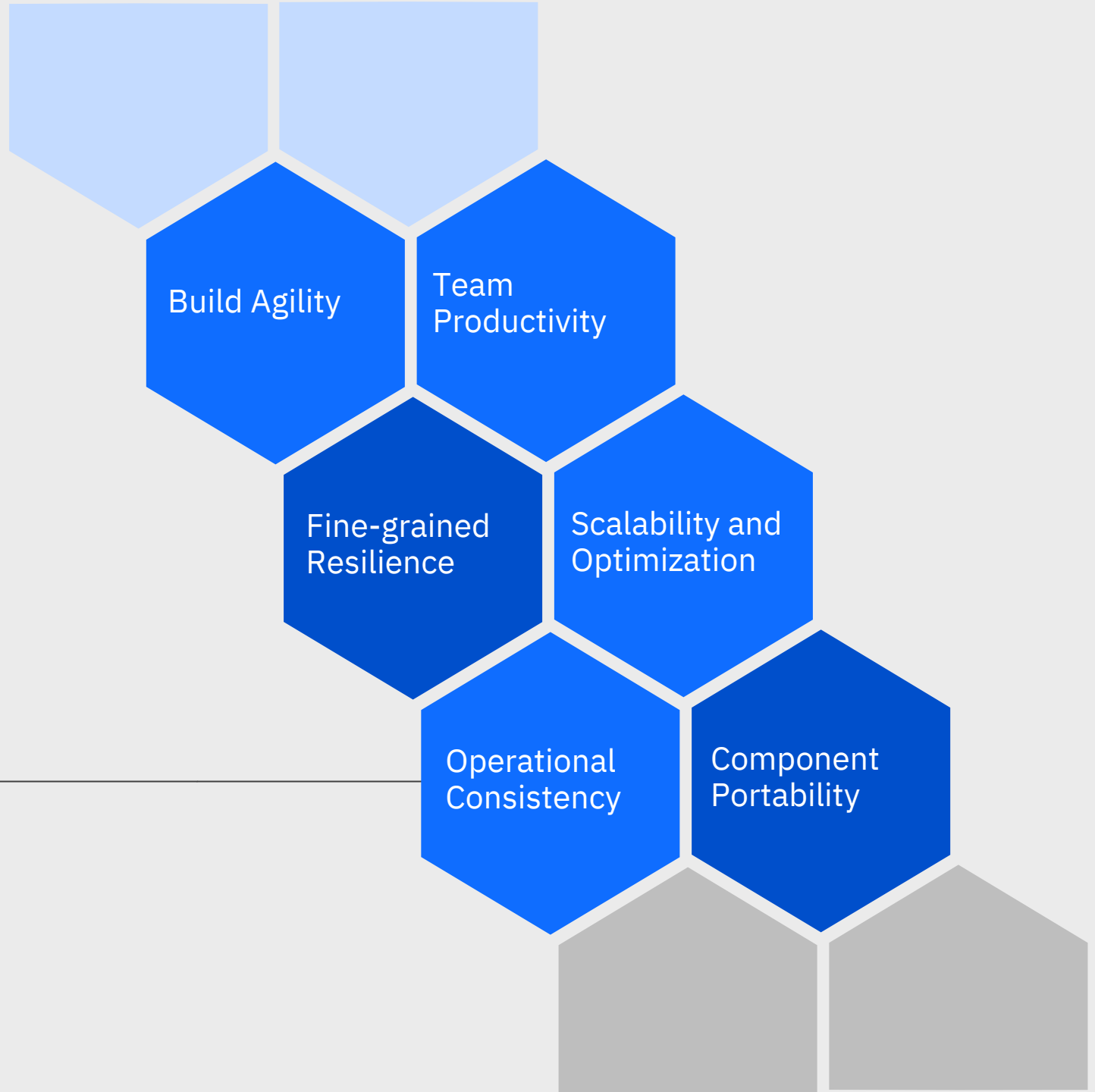
Replication:
minimum 2
maximum 2
Spread across zones
Balance workload evenly

Container platform
(declarative logical configuration)

Containers enable fine-grained deployment...and discrete scaling/availability policies with near-complete abstraction from physical resources



Benefits of a container based strategy

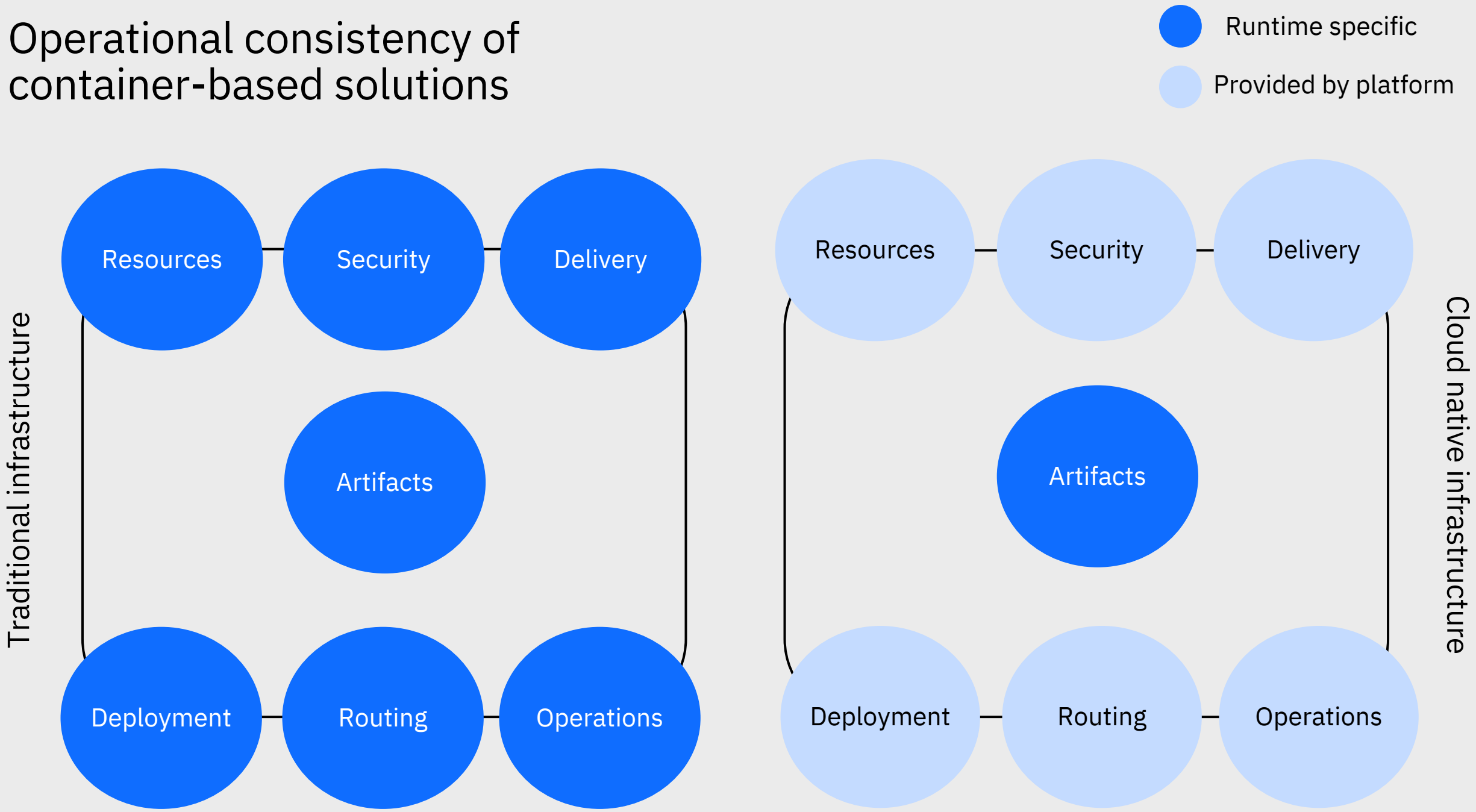


Operational Consistency

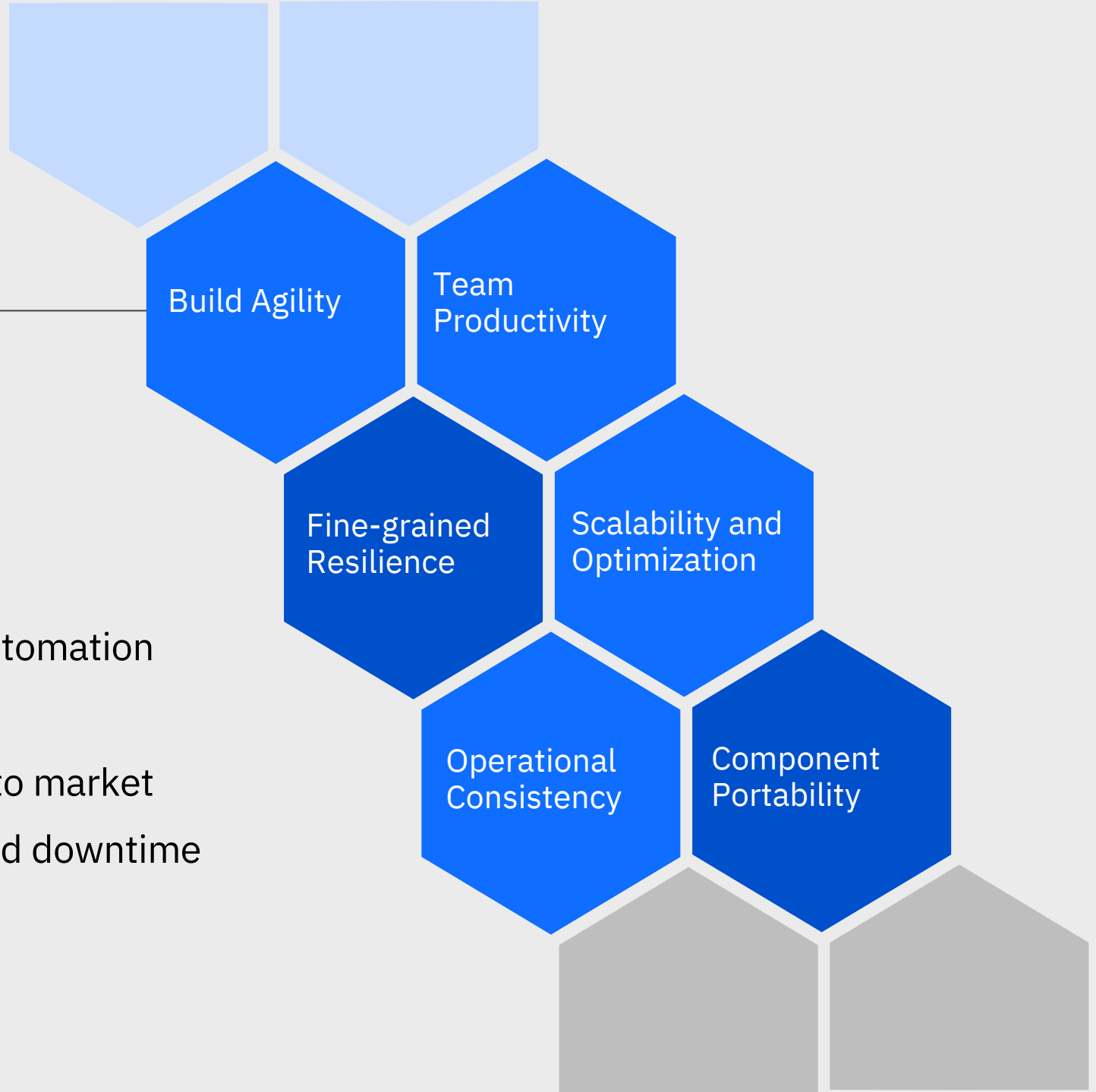
Single infrastructure skillset

Cross product consistency

Operational consistency of container-based solutions



Benefits of a container based strategy



Build Agility

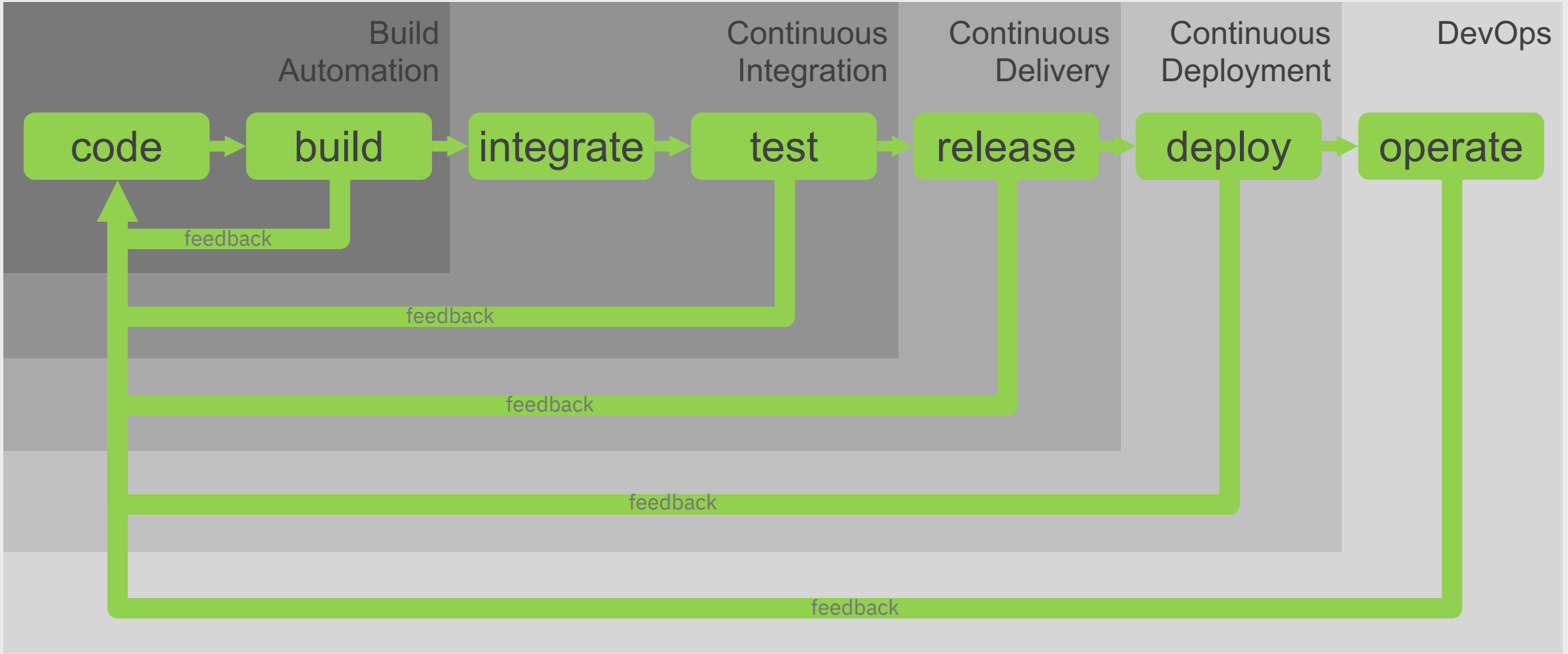
How

- Agile development processes
- Minimum-touch CI/CD pipeline automation

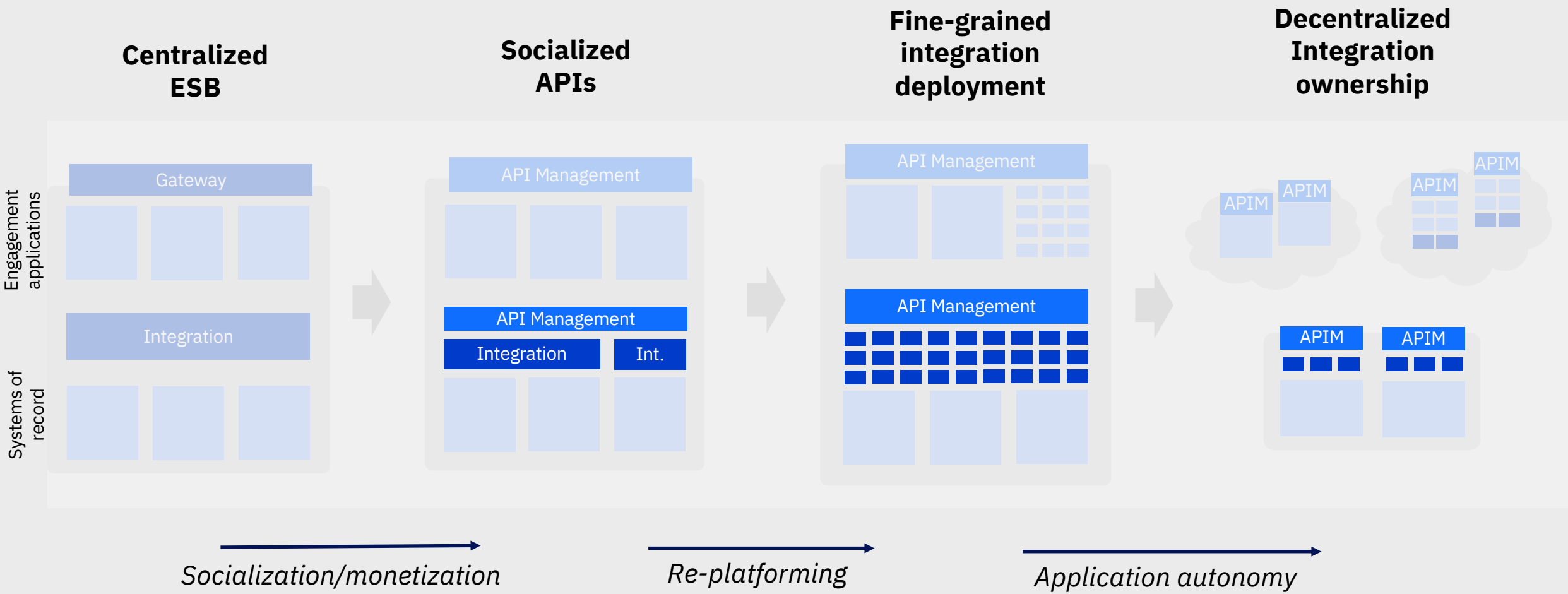
Benefits

- Fast iteration cycles - Rapid time to market
- Deployment consistency – reduced downtime

Pipeline automation (CI/CD) is fundamental groundwork for DevOps



Evolution to **agile integration** – Focus in on API Management



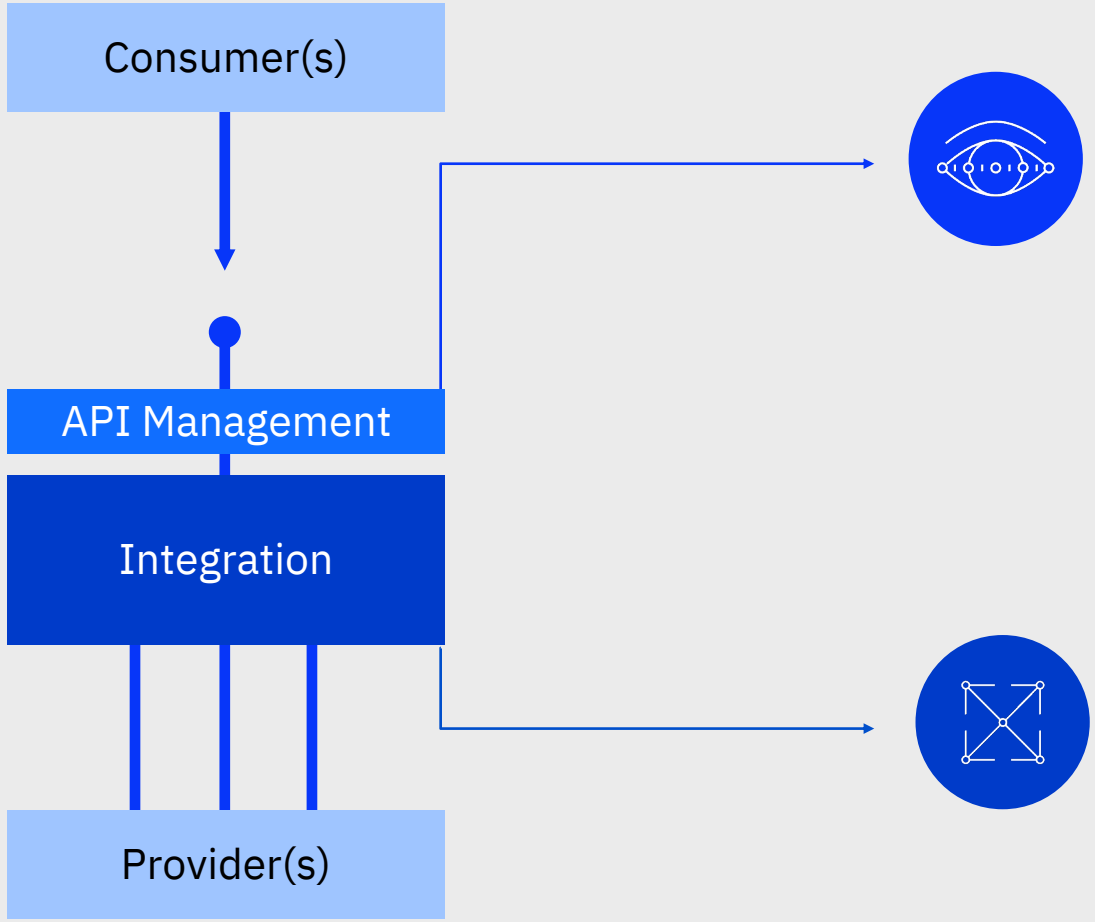
Webinars http://ibm.biz/agile-integration-webcasts

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

IBM Redbook http://ibm.biz/agile-integration-redbook

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)

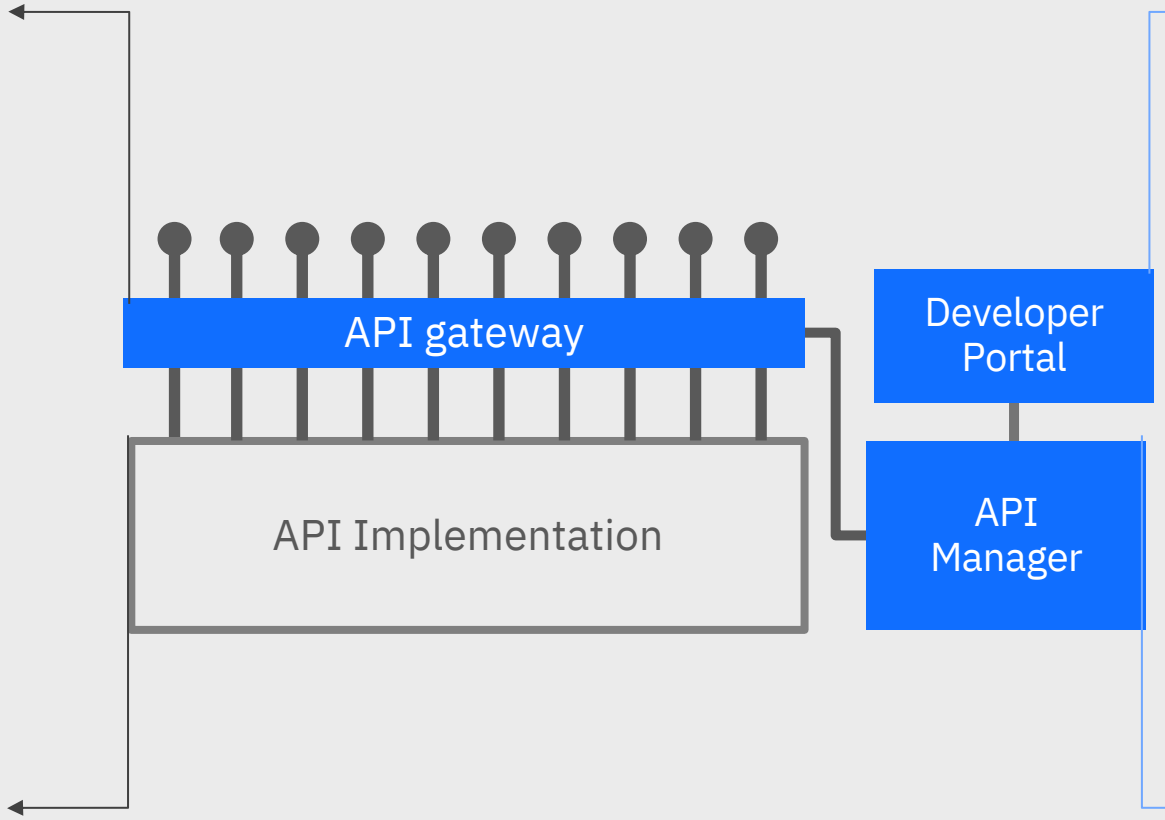
Composition: Implements the custom “integration logic”, including aggregation from multiple sources, and merging of data.

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.

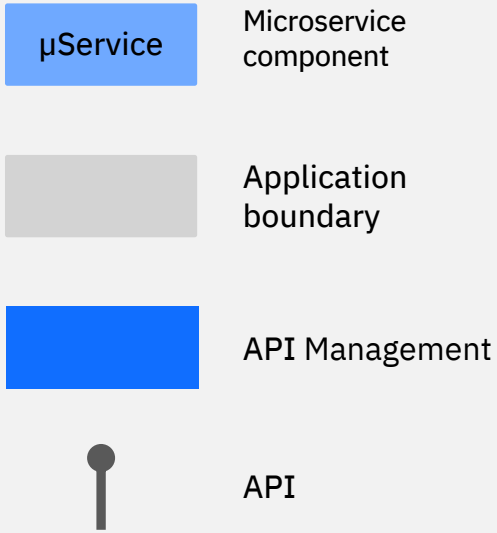


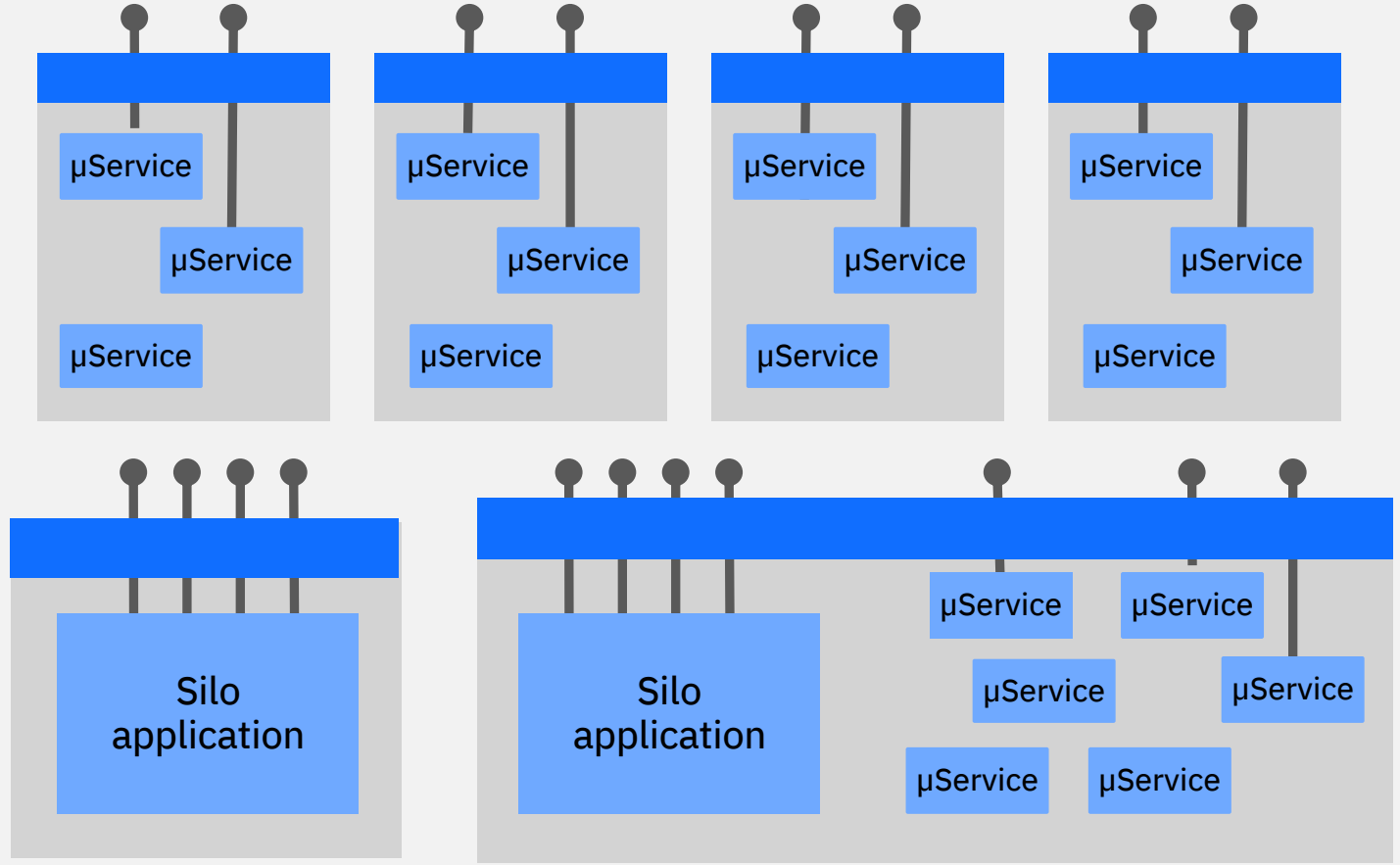
Developer portal:
- API discovery
- Self-service
  - Onboarding
  - API subscription
- Account usage analytics

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

Boundaries make complex environments manageable

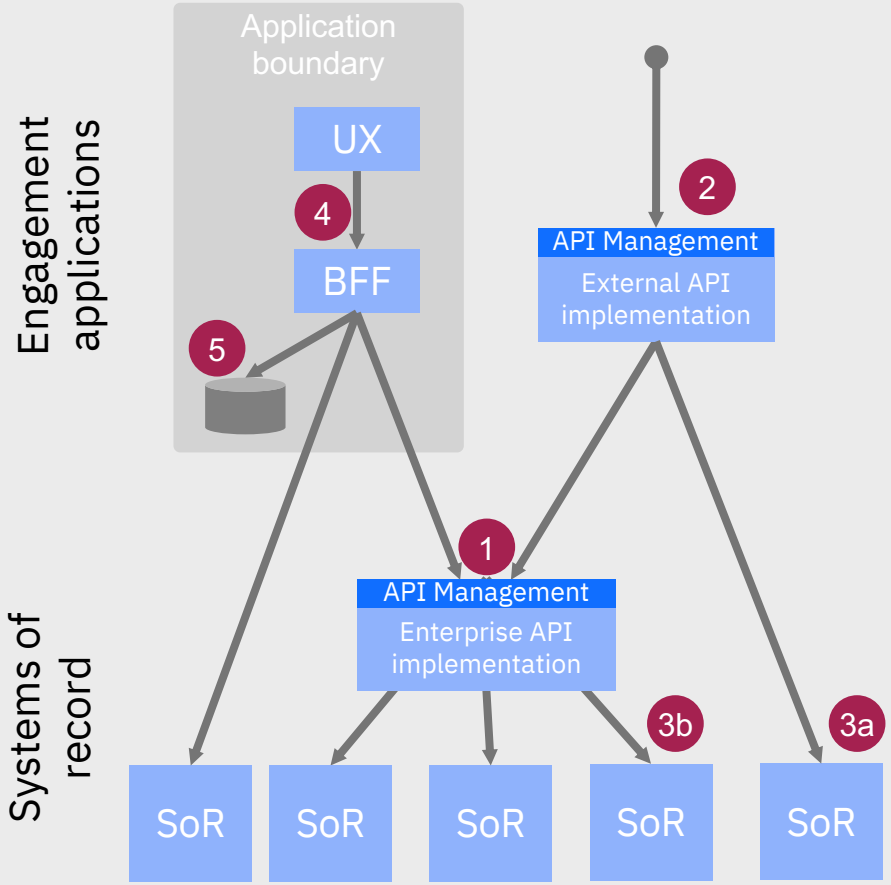
Managed API gateways define and enforce application boundaries





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

GraphQL architectural positioning: Where might it be used?

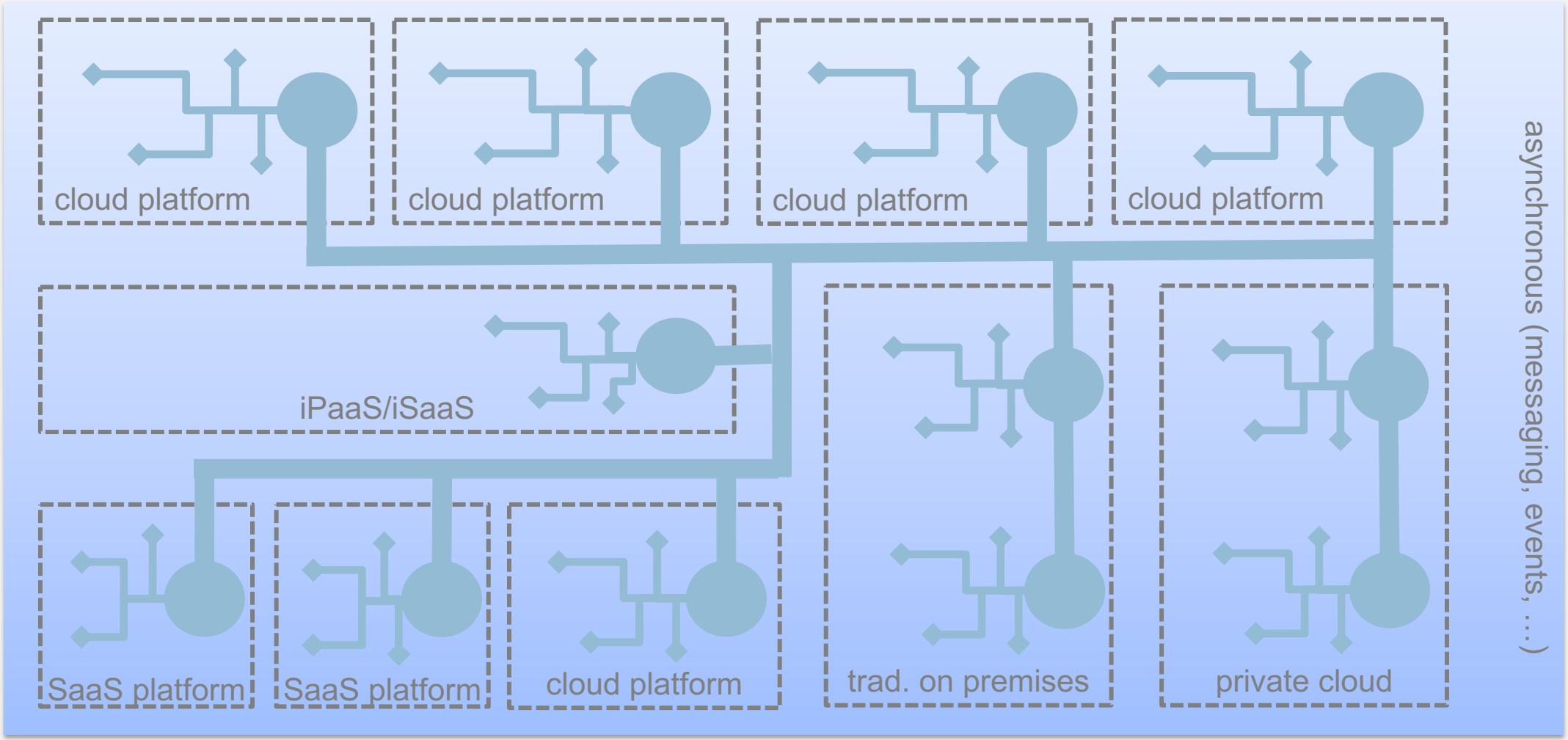


1. Enterprise wide API *
2. External API*
3. Systems of record API
4. Back end for front end API
5. Datastore API

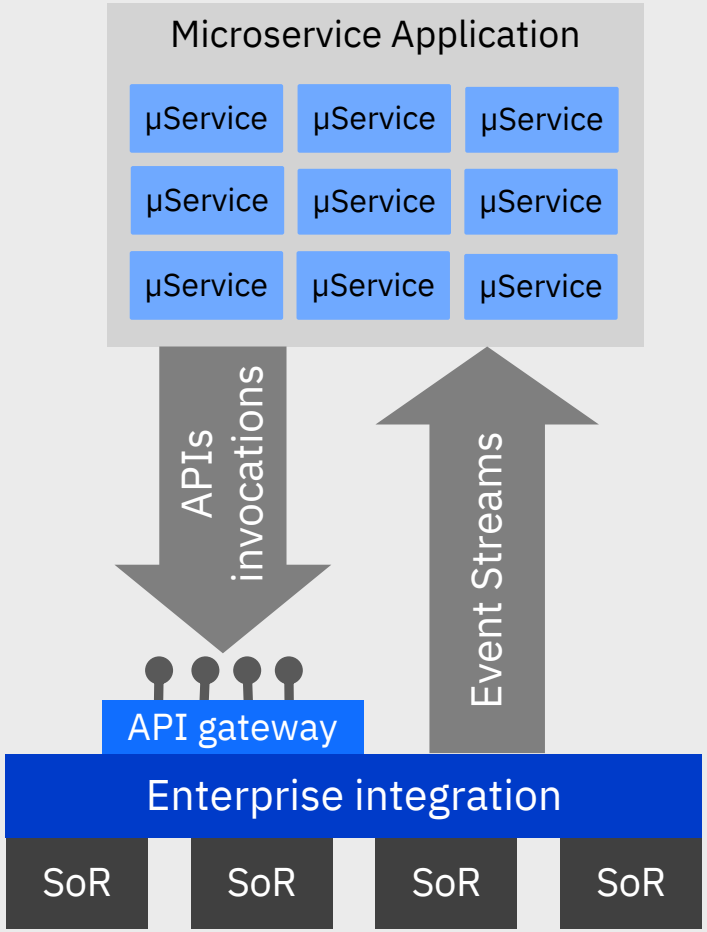
* Places where API Management may be required

The asynchronous backplane (messaging, events)

The asynchronous backplane provides reliable message/event storage **and** a distribution network that can traverse application and cloud boundaries robustly.



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

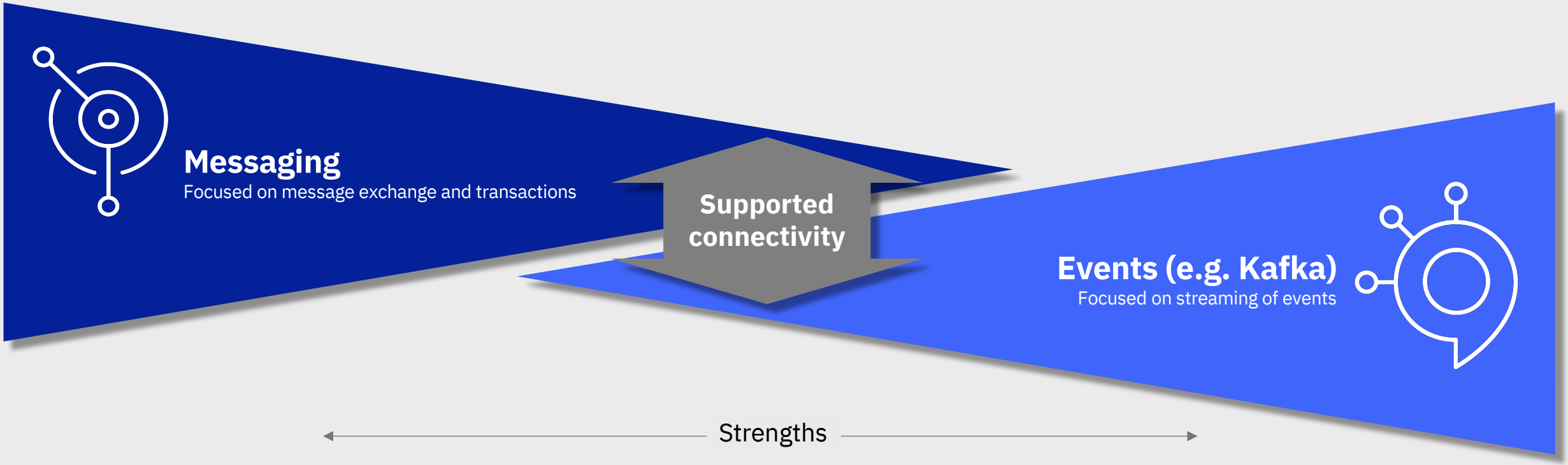
Event streams

Enable microservices to build decoupled views of the data

The best of both

Critical data exchange ← → Event driven ← → Event streaming

- **Assured Delivery**
- **Fine grained messaging**
- **Topics and Subscriptions**
- **Scalable Subscription**
- **Stream History**



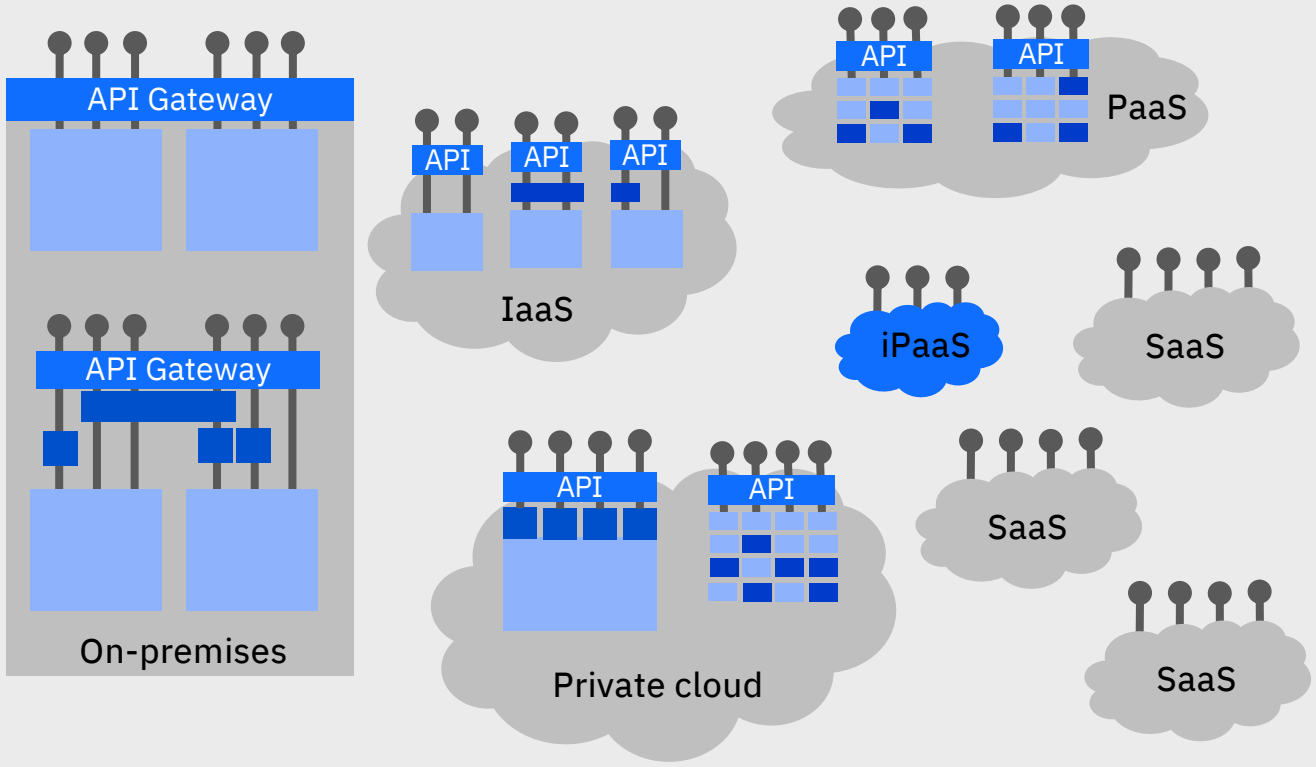
Strengths

A realistic look at the proportions of integration deployment

Centralized integration will still be critical for complex or highly governed integrations, and to maintain the existing integrations that have no need to modernize. It may or may not move to containers depending on needs for high agility, elastic scalability, discrete resilience etc.

Many applications will lift and shift to infrastructure as a service (IaaS) due to the high cost of refactoring. Again, their integrations may remain centrally maintained and operated in order to reduce operational and procedural change.

Convergence of application and integration is primarily within the space of new application development.



IBM Cloud Pak for Integration

Broadest integration capabilities

Unified experience, operational efficiency & reuse

–

Deploy where needed

Cloud ready architecture designed to maximize a range of industry leading platforms and clouds

–

Enterprise-grade

Secure, scalable, battle hardened

API Lifecycle

Application & Data Integration

Enterprise Messaging

Events

High Speed Transfer

Integration Security

Large UK based, global retailer

Issues

- Provisioning of new integration infrastructure taking weeks to months
- Centralized Infrastructure, maintenance slots for upgrades of integration infrastructure nearly impossible to co-ordinate between app teams
- Centralized CoE , leading to bottlenecks in introduction of new integrations

Pursuing cloud first to cloud only digital transformation for rapid innovation and cost optimization.

Championed *agile integration* supported by IBM Cloud Pak for Integration

- **Rapid provisioning**: Automated deployment enables provisioning of integration infrastructure and new pattern-based integration in minutes
- **Evergreening**: Continuous adoption of integration software updates ensures latest features and security patches always in place with each deployment
- **Productivity**: Pattern-based integrations with automated build and deployment to empower digital teams of non-integration specialists
- **Fine-grained control**: Integrations deployed in full isolation through containerization, providing independent agility, scalability and resilience
- **Decoupling**: Event streams and Messaging serve up data from back end systems in real-time enabling APIs and applications to build local caches of data
- **Hybrid connectivity:** A stable and capable on-premise integration runtime environment enabling seamless connectivity between on-premise and cloud apps

Benefits of **Agile Integration**

Productive workforce

- Empowered autonomous teams
- Faster delivery: shorter iterations, automated pipeline
- Increased innovation, reduced fear of change
- Clear fine-grained component ownership
- Consistent skillset across multiple capabilities

Robust solutions

- Implicit high availability
- Reversible deployments
- Fine-grained resilience
- Elastic scalability
- Consistency across environments

Efficient provisioning

- Rapid provisioning of integration capabilities
- Optimized use of software and hardware
- Evergreen, always current software runtimes
- Granular security and access control
- Implicit networking

Learn more

Integration Modernization

Agile Integration
ibm.com/cloud/integration/agile-integration

Ovum Analyst Report: Hybrid Integration Platforms
ibm.biz/ovum-hip

IBM Cloud Pak for Integration

Product Page
ibm.com/cloud/cloud-pak-for-integration

Get Started
ibm.com/cloud/cloud-pak-for-integration/get-started

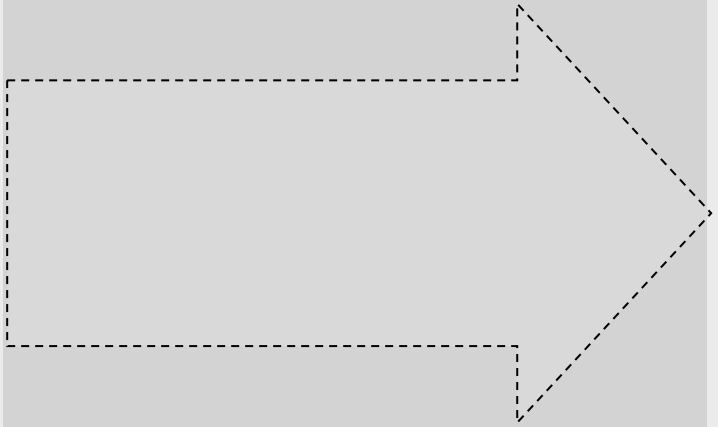
Traditional Integration

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

Thank you!