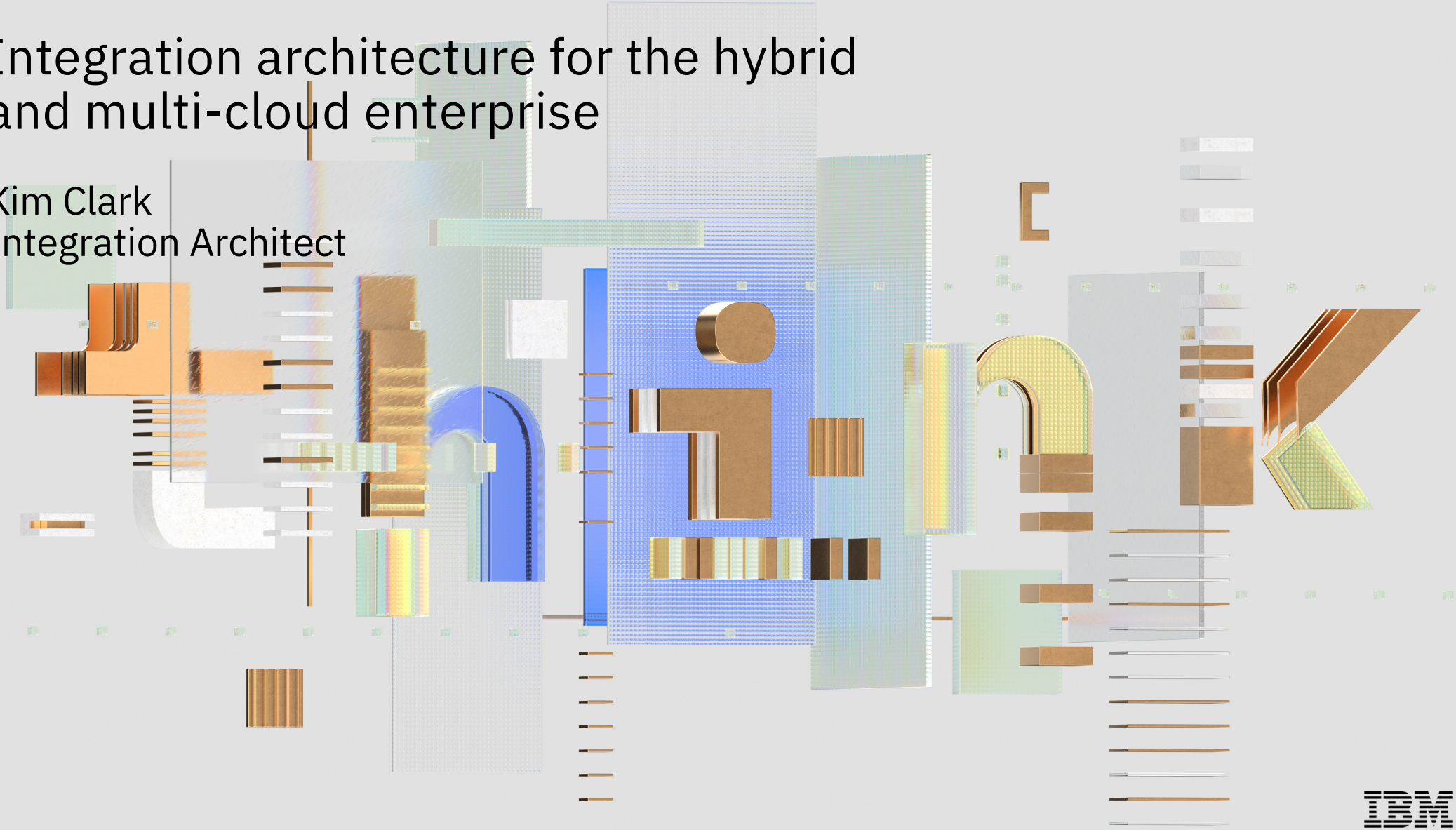
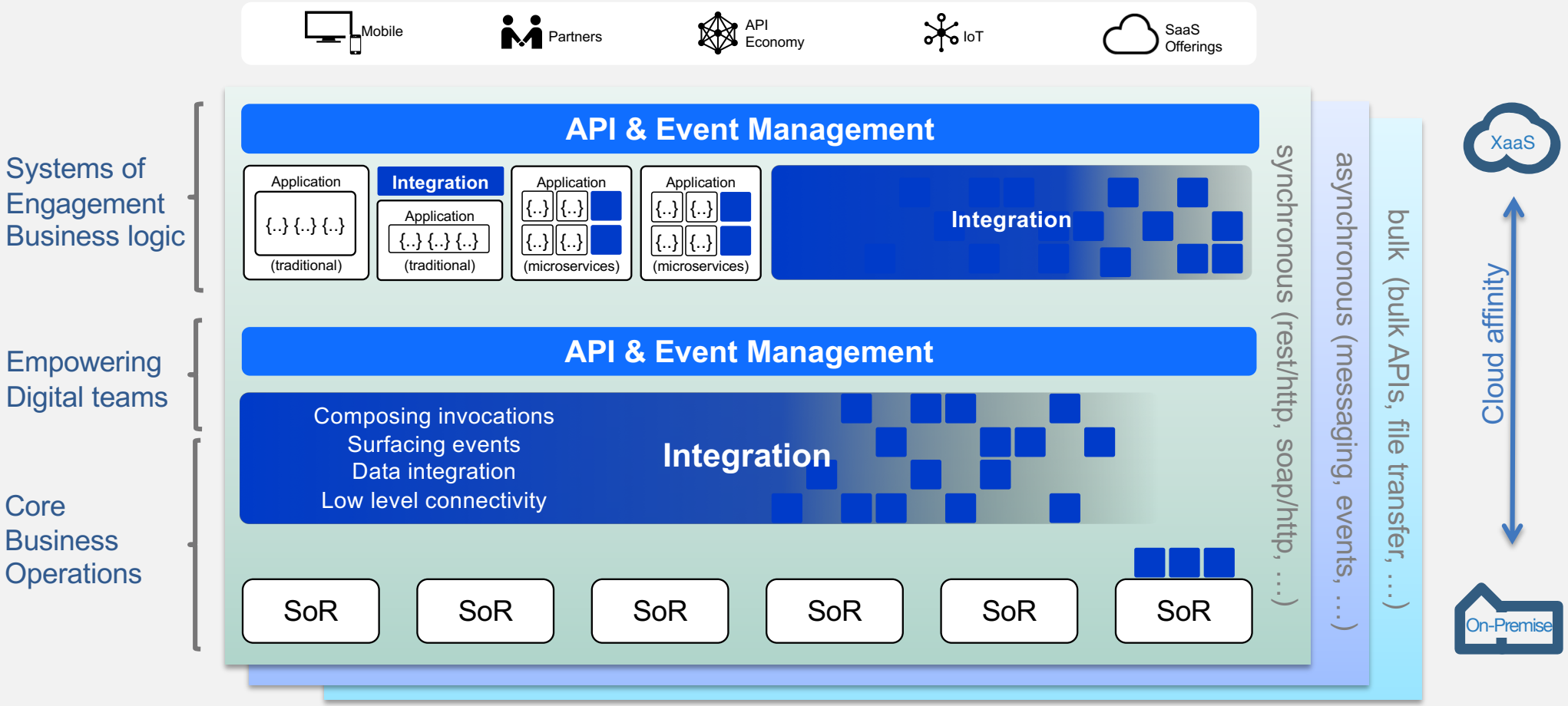
Integration architecture for the hybrid and multi-cloud enterprise

Kim Clark
Integration Architect



IBM

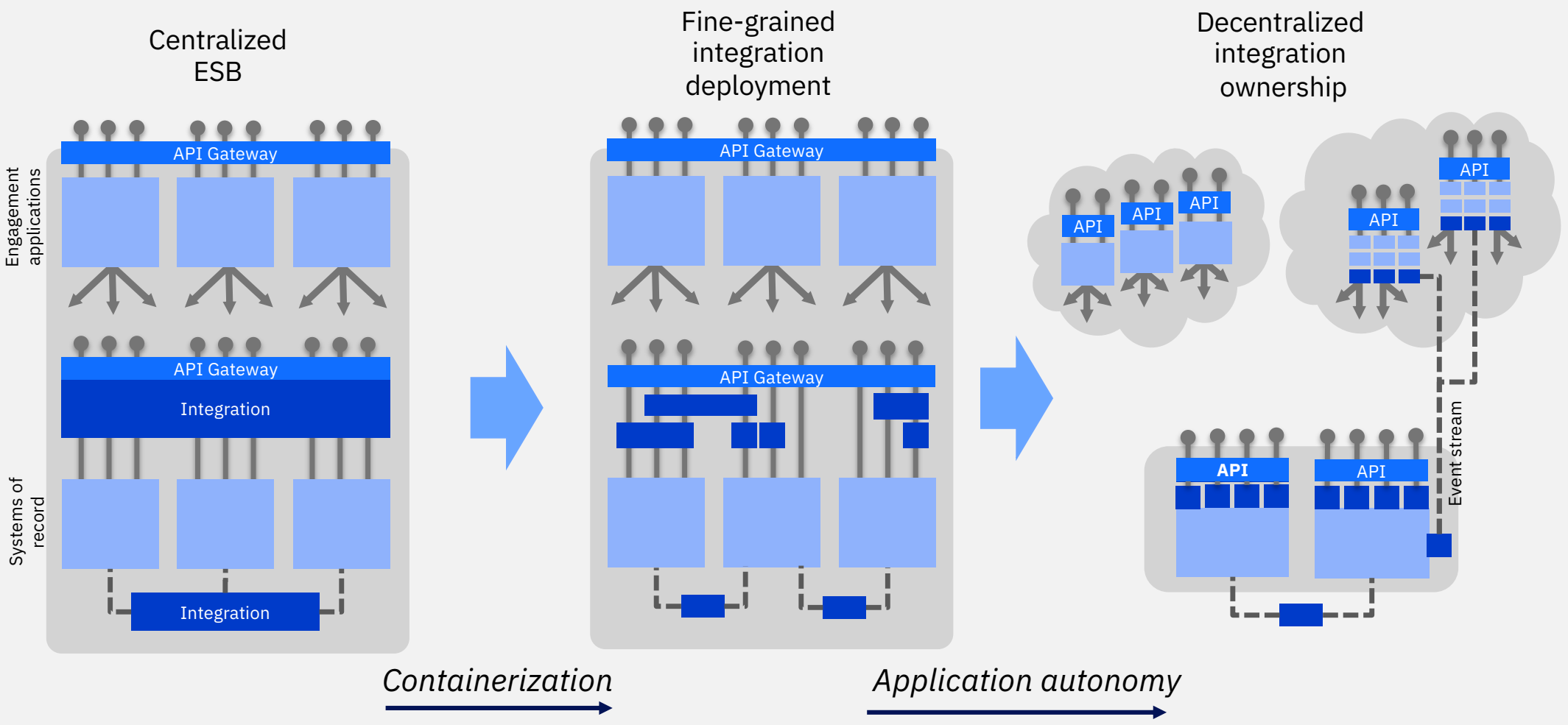
Integration Reference Architecture



webinar http://ibm.biz/HybridIntRefArchYouTube

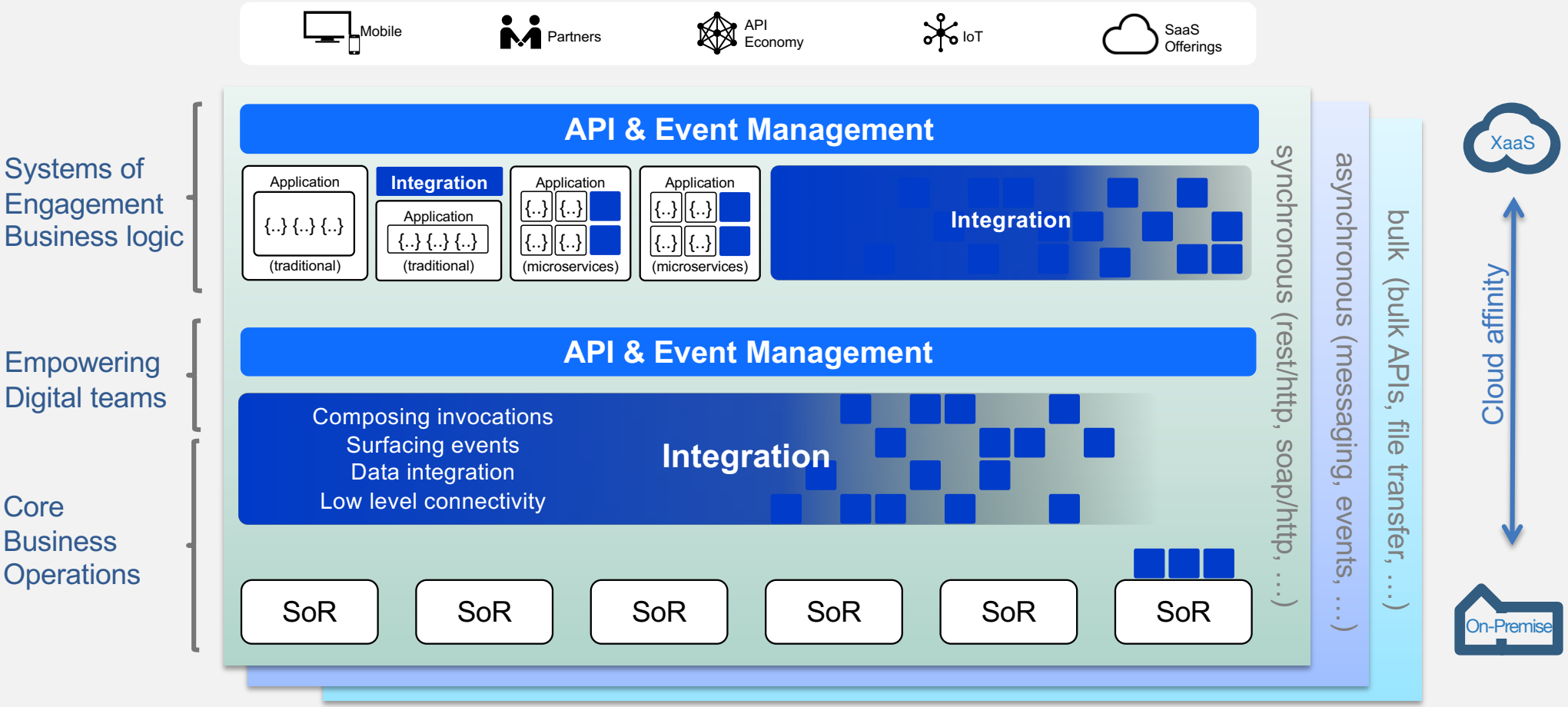
paper http://ibm.biz/HybridIntRefArch

Evolution to **agile integration architecture**



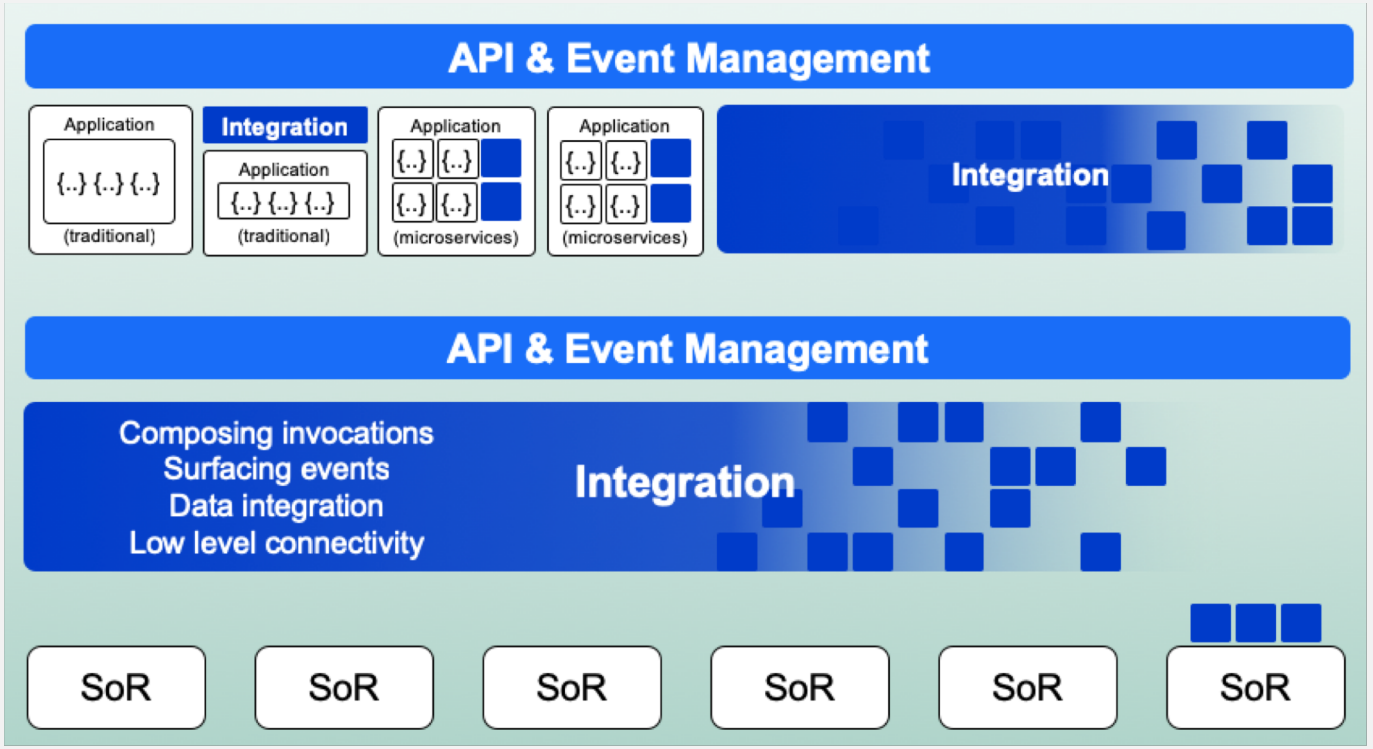
Webinar http://ibm.biz/AgileIntegArchWebinar
eBooklet https://www.ibm.com/cloud/agile-integration-architecture

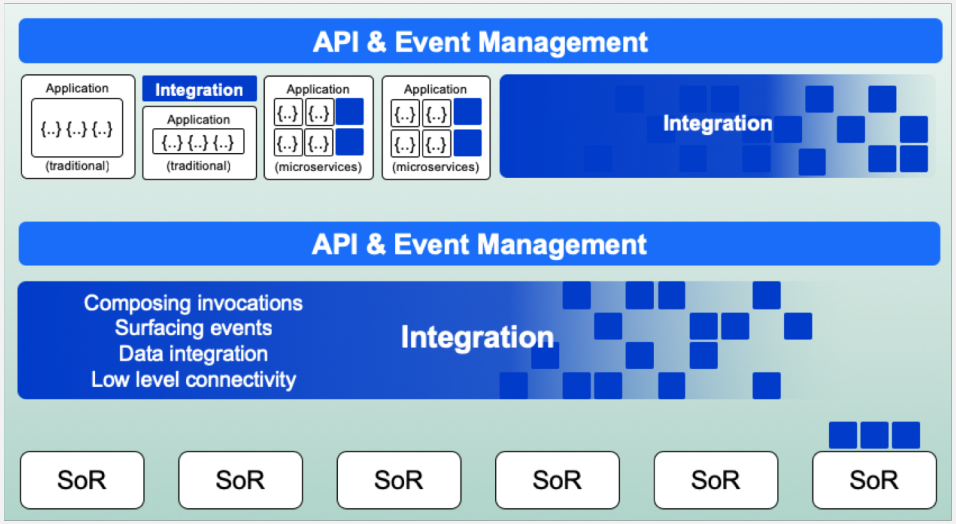
Integration Reference Architecture



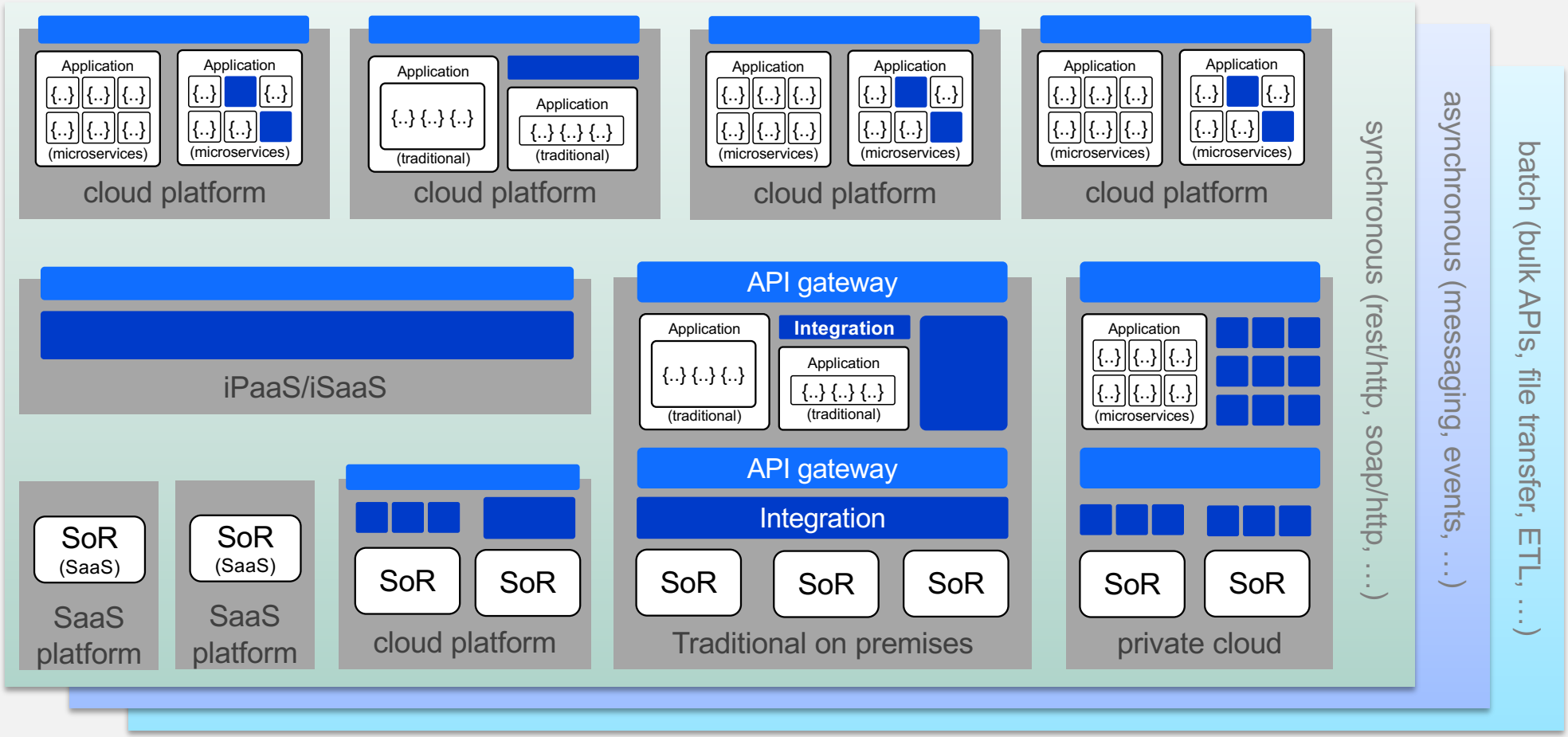
webinar http://ibm.biz/HybridIntRefArchYouTube

paper http://ibm.biz/HybridIntRefArch

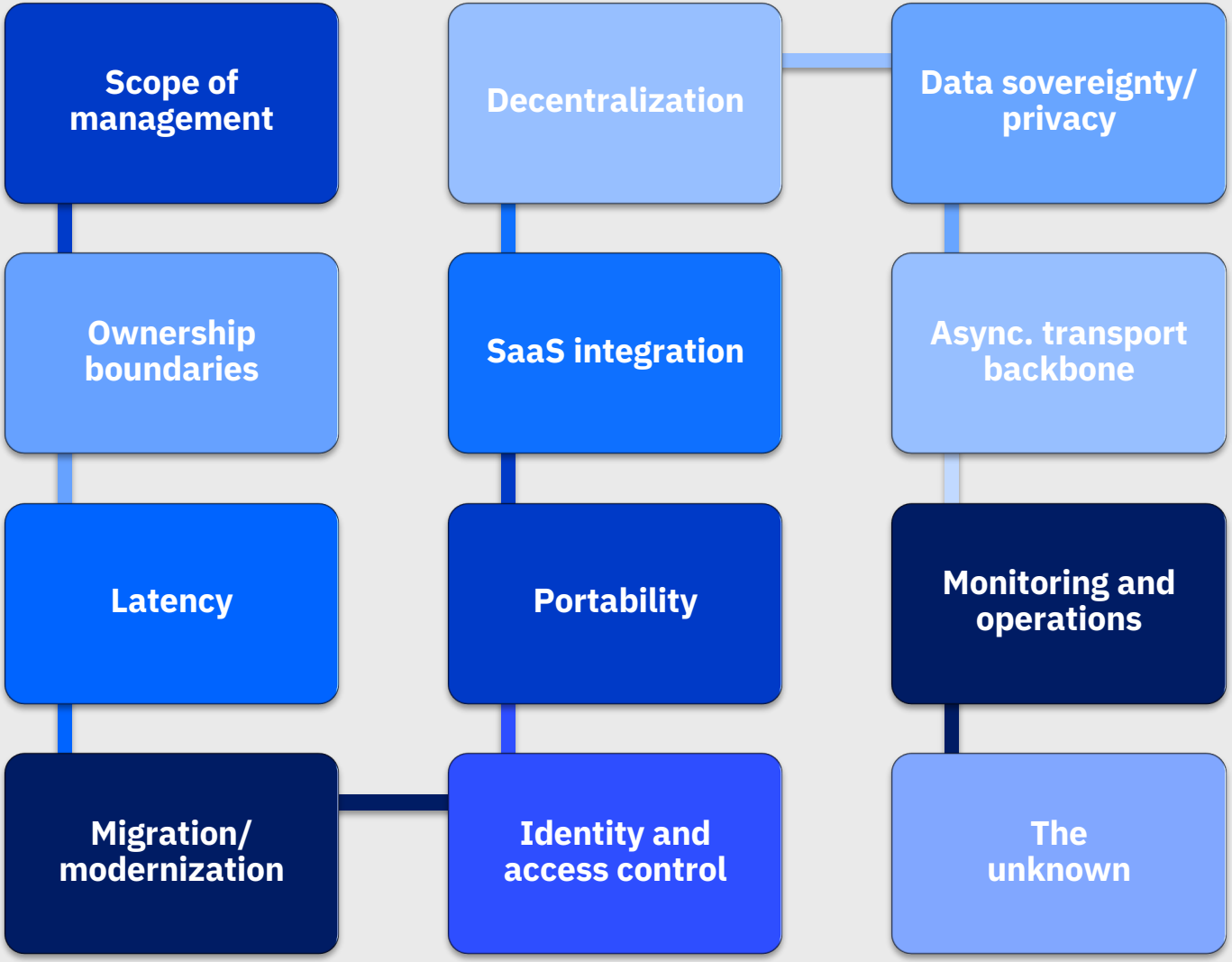

API & Event Management
Application
{..} {..} {..}
(traditional)
Integration
Application
{..} {..} {..}
(traditional)
Application
{..} {..}
{..} {..}
(microservices)
Application
{..} {..}
{..} {..}
(microservices)
Integration
API & Event Management
Composing invocations
Surfacing events
Data integration
Low level connectivity
Integration
SoR
SoR
SoR
SoR
SoR
SoR


API & Event Management
Application
{..} {..} {..}
(traditional)
Integration
Application
{..} {..} {..}
(traditional)
Application
{..} {..}
{..} {..}
(microservices)
Application
{..} {..}
{..} {..}
(microservices)
Integration
API & Event Management
Composing invocations
Surfacing events
Data integration
Low level connectivity
Integration
SoR
SoR
SoR
SoR
SoR
SoR

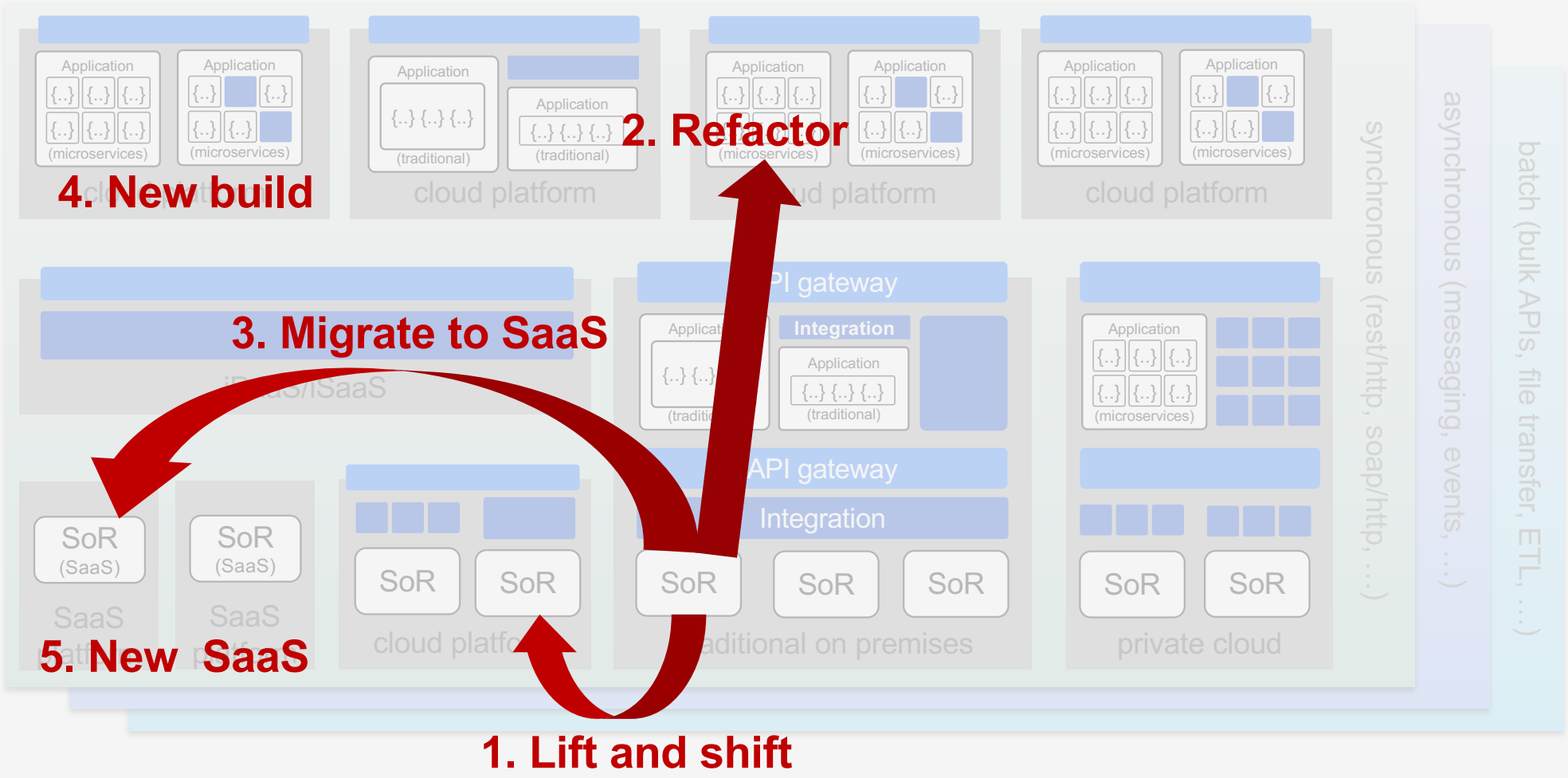
The multi-cloud challenge



What are some of the multi-cloud challenges?



Common scenarios leading to multi-cloud

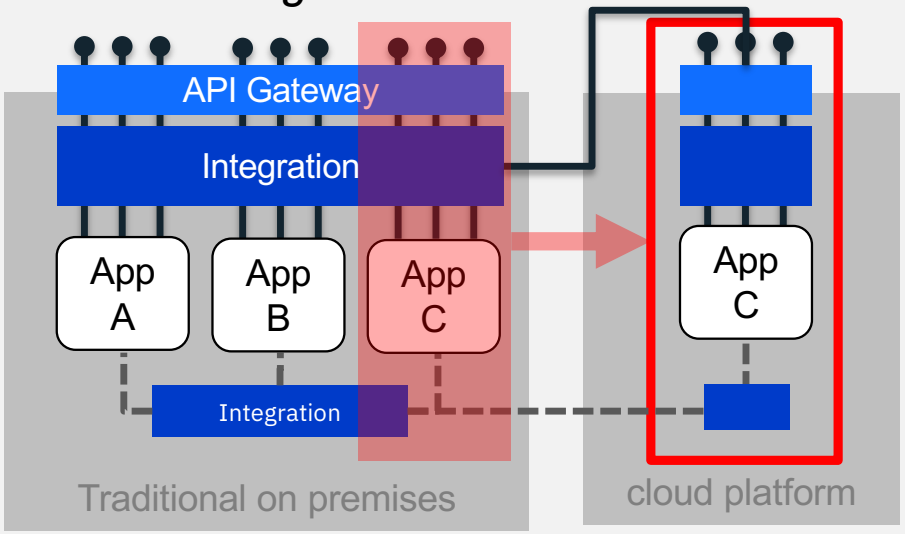


Scenario 1: ***Lift and shifting an application to the cloud***

What? Application moved onto cloud infrastructure with minimal change
Why? You no longer need to own/run the datacenter

Integration specific challenges

- Do the integrations need to remain close to the SoR?
- When can you safely move the exposure point for APIs?
- How will cross application synchronous requests be exposed?
- How will cross-cloud asynchronous communication be enabled?
- If there is cross-application integration who does it belong to?
- Can/should the integration be refactored/optimized?



How can integration modernization help?

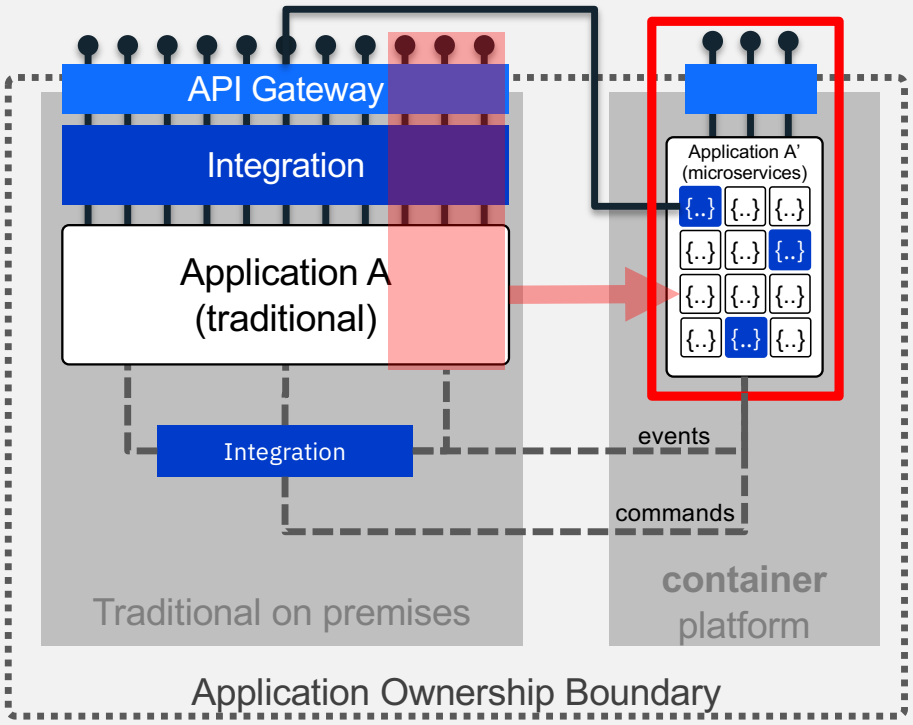
- Isolated integrations
- Infrastructure agnostic
- Application owned integrations
- Better integration partitioning
- Clearer application API boundaries
- Multi-cloud enabled asynchronous transport
- Lower risk, higher velocity refactoring

Scenario 2: ***Refactor for application modernization (e.g. strangler pattern)***

What? Portions of an application are progressively refactored
Why? Agility for the parts of the application that matter

How is this different from Scenario 1?

1. *The application is being **refactored***
2. *A **single** application is spread across a multi-cloud landscape*



Integration specific challenges

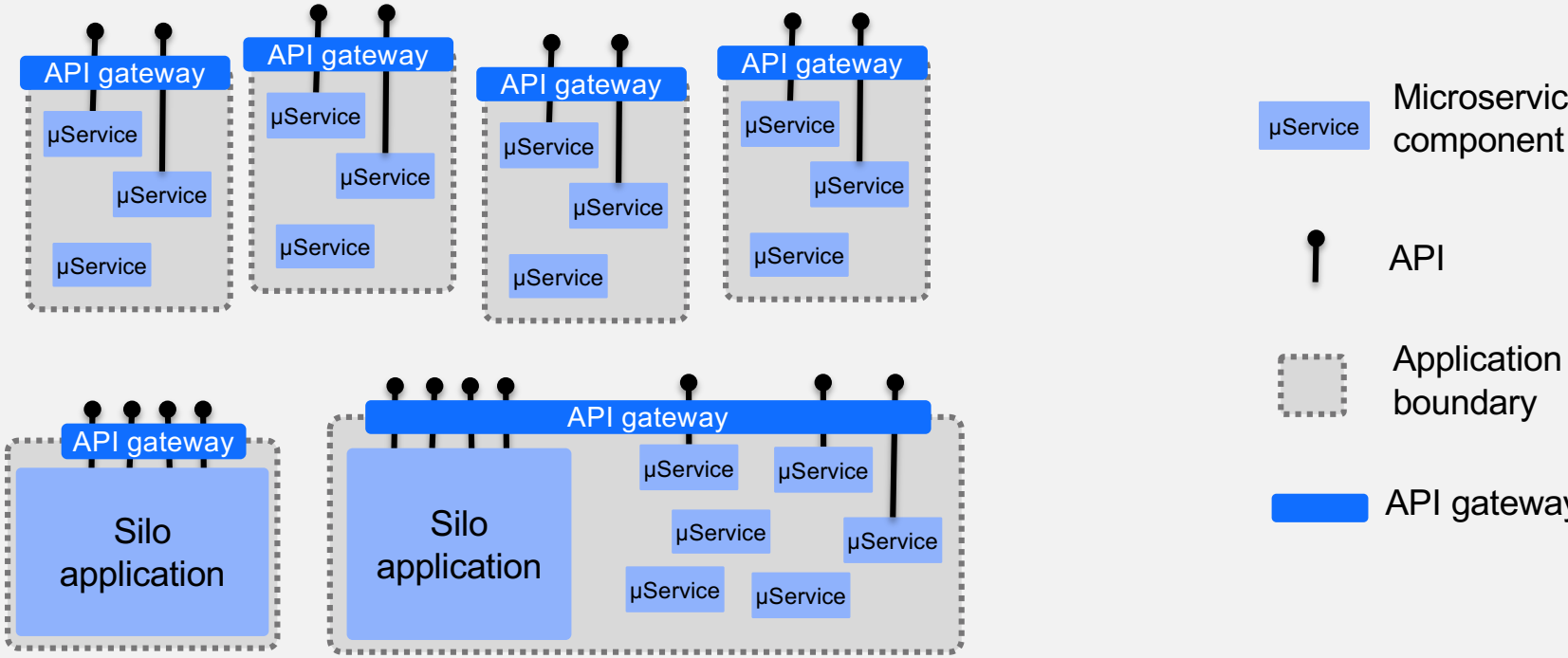
- Function calls that used to be local are now remote
- How do we manage the re-location of API endpoints
- How do we rapidly make data available as event streams
- Transactional updates are now across distributed data

How can integration modernization help?

- API management (+ integration) to provide secure private access to application internal functionality
- Federated gateways with centralized management to enable single point of control for all application APIs
- Integration can hook into existing applications to propagate streams for data change events
- Transactional messaging can provide assured, once only delivery of commands for eventual consistency
- A lightweight integration runtime may be a good candidate for some of the new microservice components

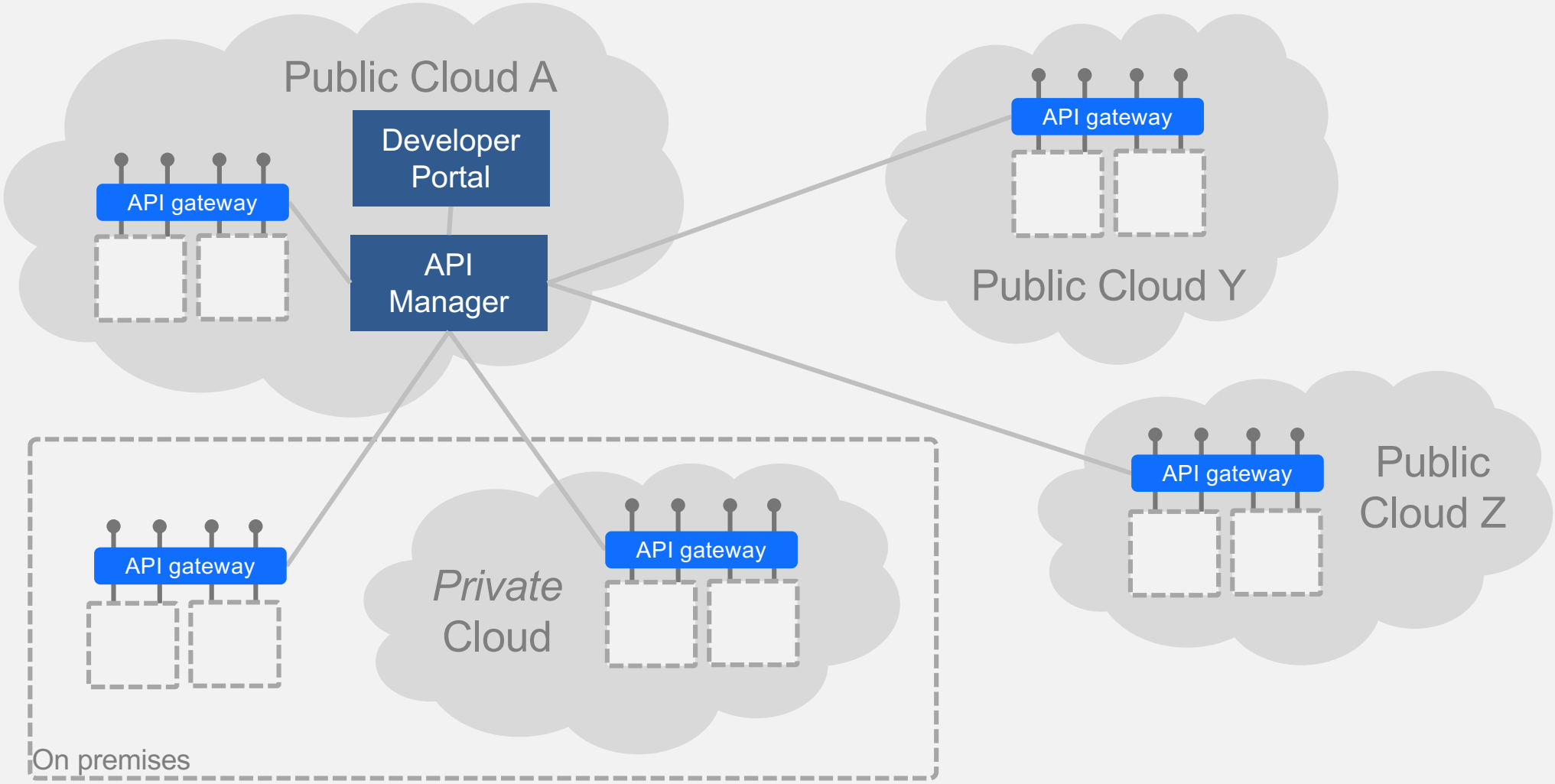
Boundaries make complex environments manageable

Managed API gateways *define and enforce application boundaries*



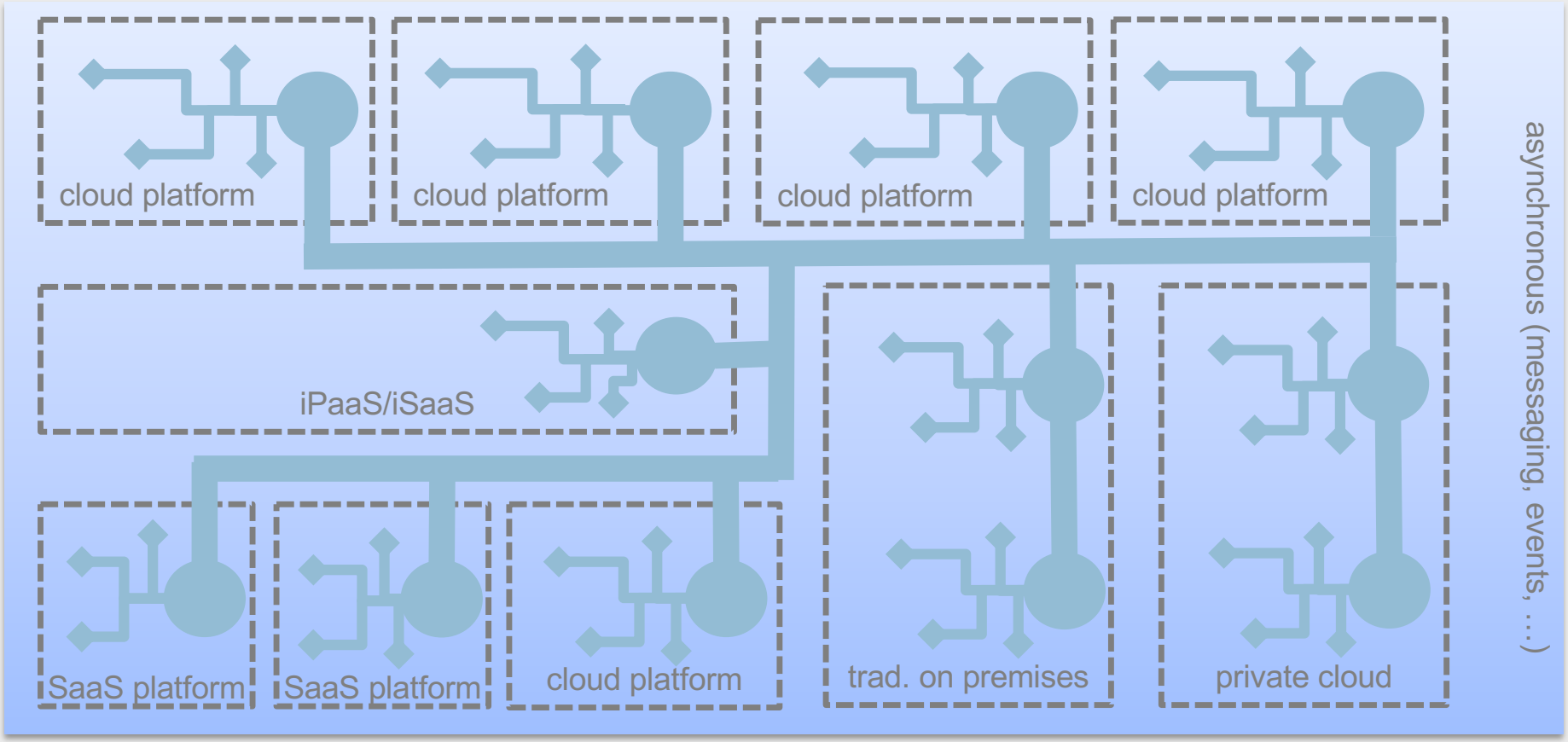
https://developer.ibm.com/apiconnect/2018/10/09/apis-microservices-defining-boundaries

Federated API gateways, centralized management



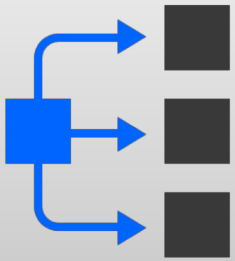
The asynchronous backplane (messaging, events)

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



Events
(notifications)

Stream History

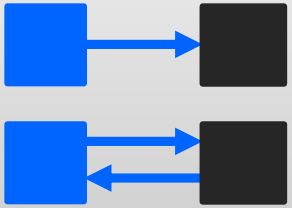


Scalable Consumption

Immutable Data

Messaging
(commands)

Transient Data Persistence



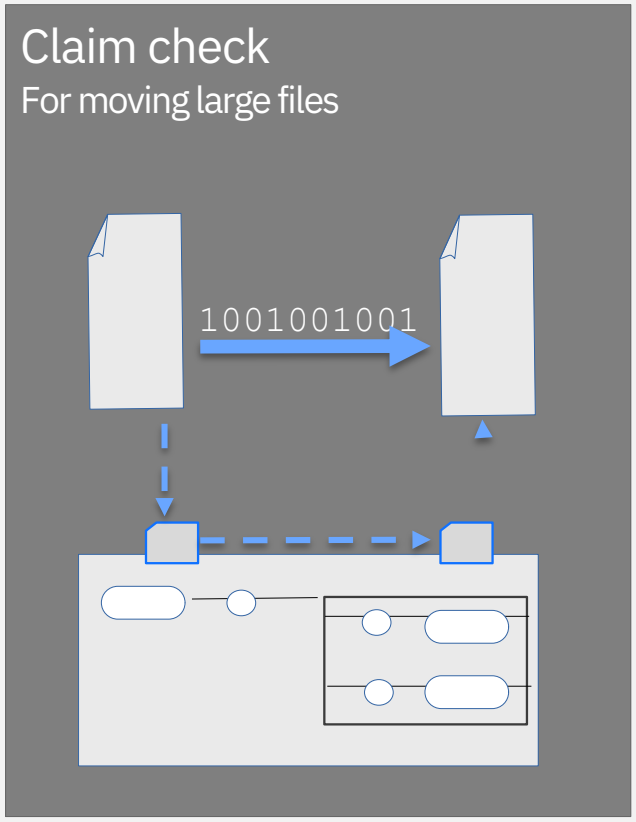
Source / Target Request / Reply

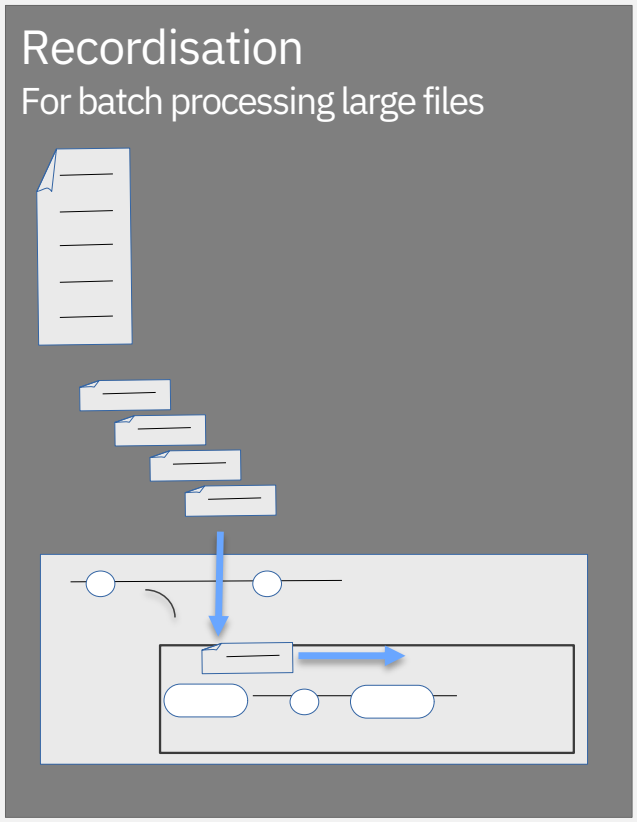
Targeted Reliable Delivery

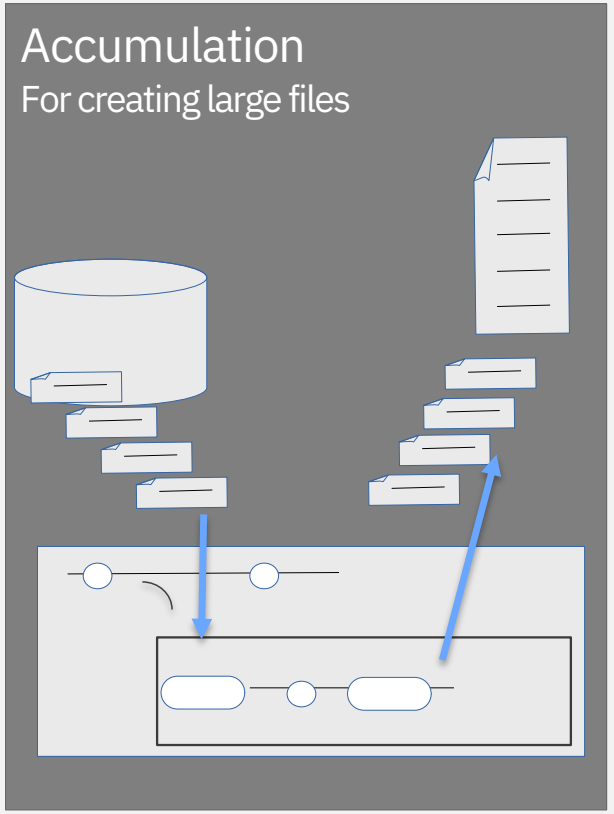
https://developer.ibm.com/messaging/2018/05/18/comparing-messaging-event-streaming-use-cases

Large file processing patterns become more critical in multi-cloud

As network separation increases, and movement of large content becomes more common, patterns that recognize the associated file movement challenges return to popularity.

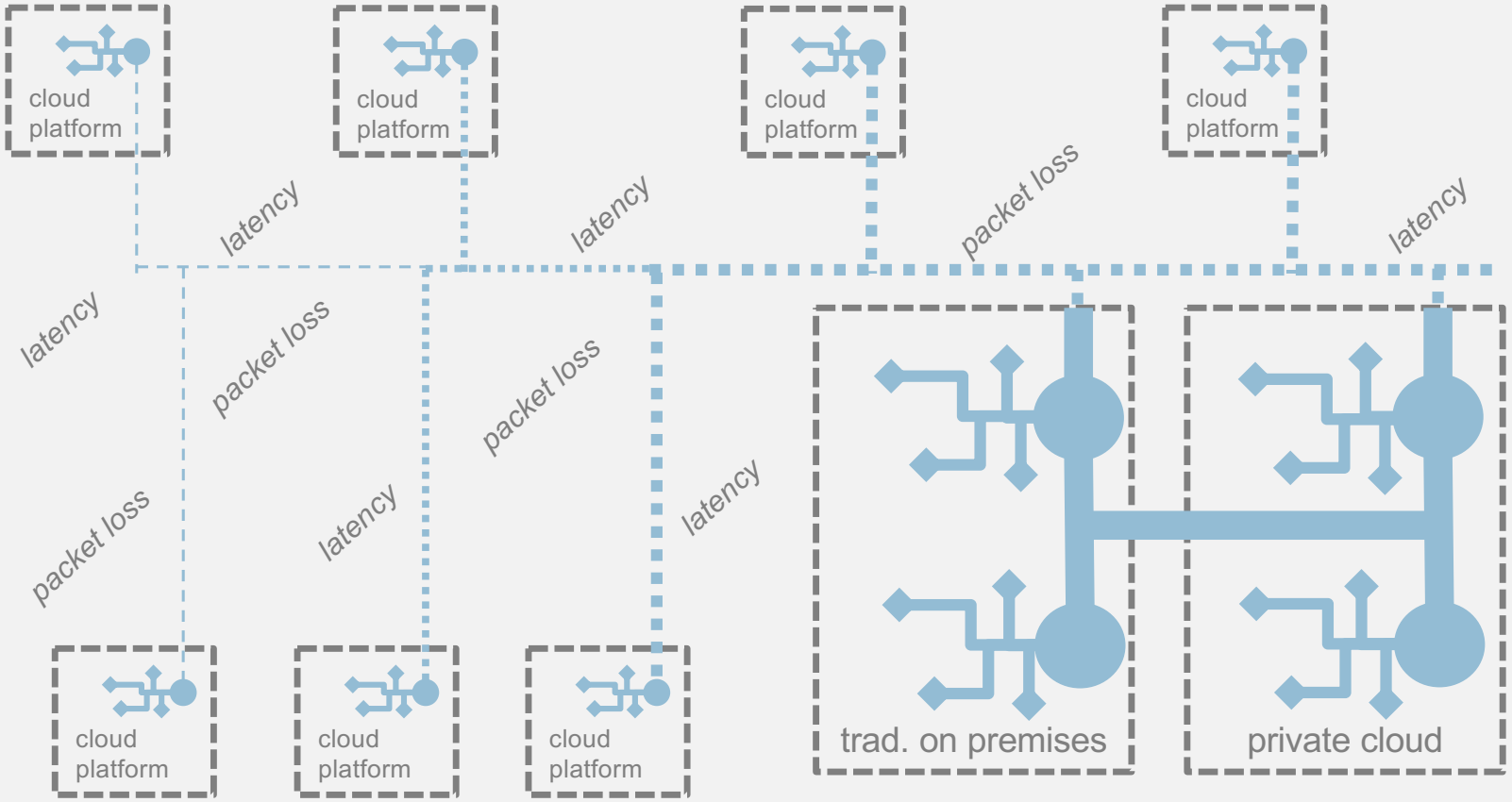






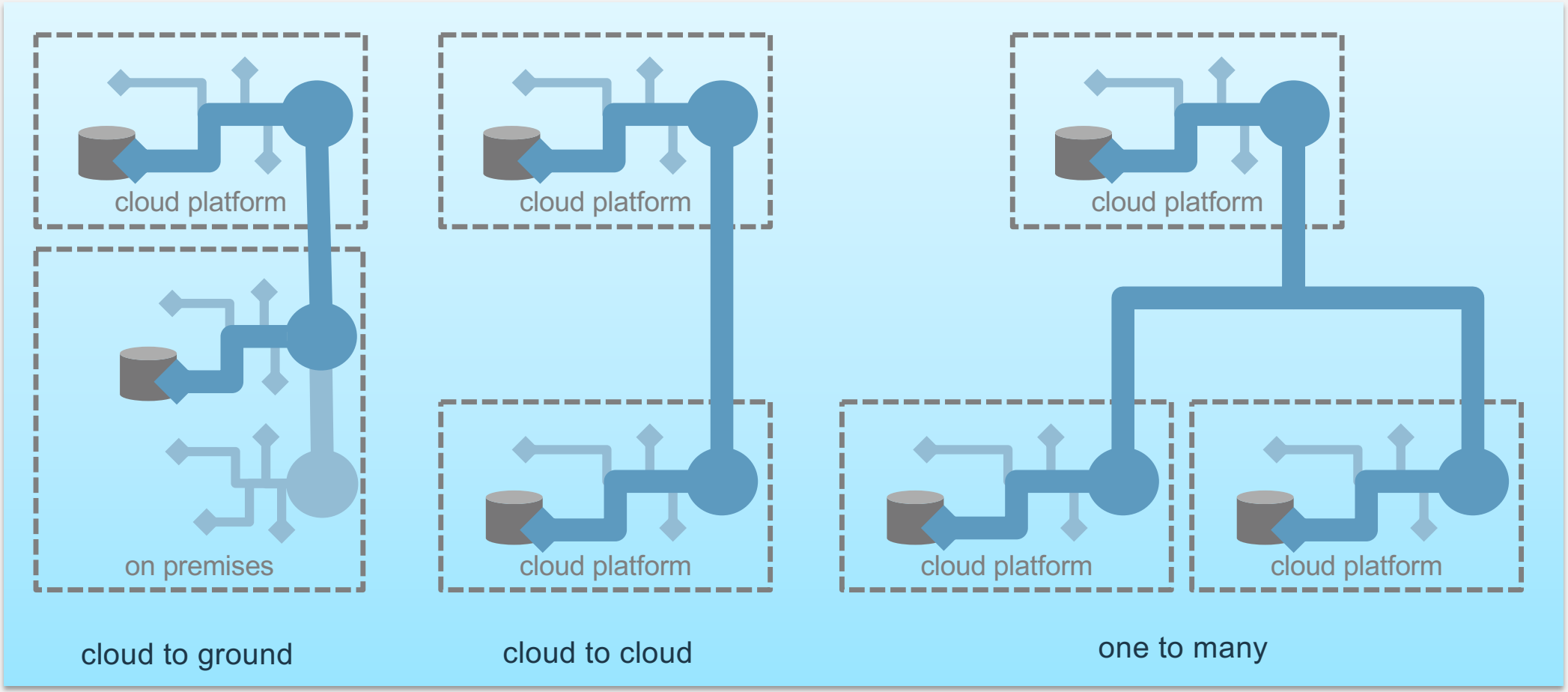
The TCP challenge for large file transfers

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.



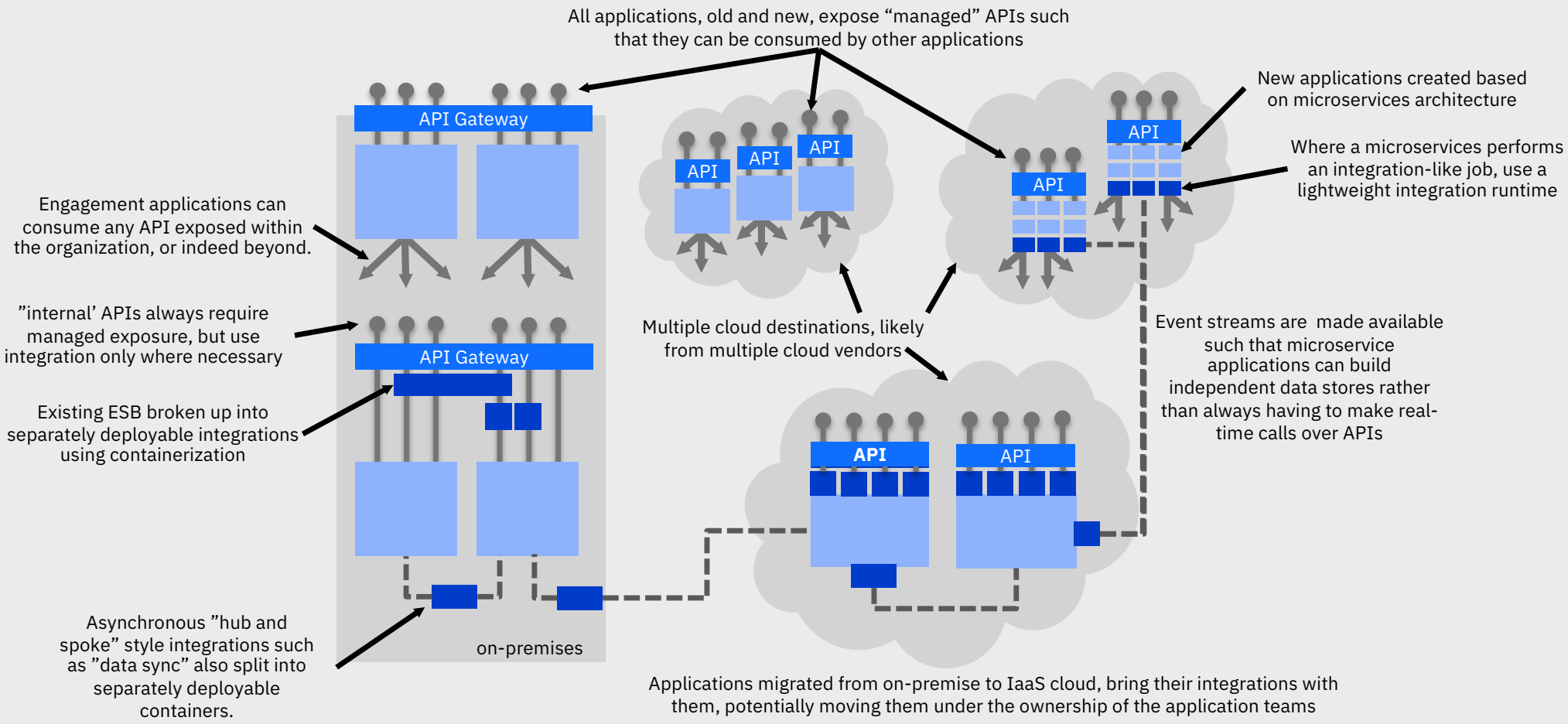
The batch/bulk backplane (file transfer, bulk APIs, …)

The batch/bulk backplane must enable end-to-end, reliable, secure, scalable, interruptible, linear transfer rates across disparate locations, regardless of geographical or network distance. It should get the data all the way to the last mile using the most efficient final protocol into the target (e.g. database, Hadoop, block storage, object storage, filestore, …)



Moving to agile integration

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



Public material on integration modernization

Agile Integration Architecture (AIA)

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

Webinar series
http://ibm.biz/AgileIntegArchPodcasts

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

Hybrid Integration Reference Architecture

https://ibm.biz/HybridIntRefArch
http://ibm.biz/HybridIntRefArchYouTube
http://ibm.biz/MultiCloudIntegrationArchitectureWebinar

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

Thank you