Experiments in AEM Author Scalability

Matt Ryan | Sr. Software Engineer | Adobe

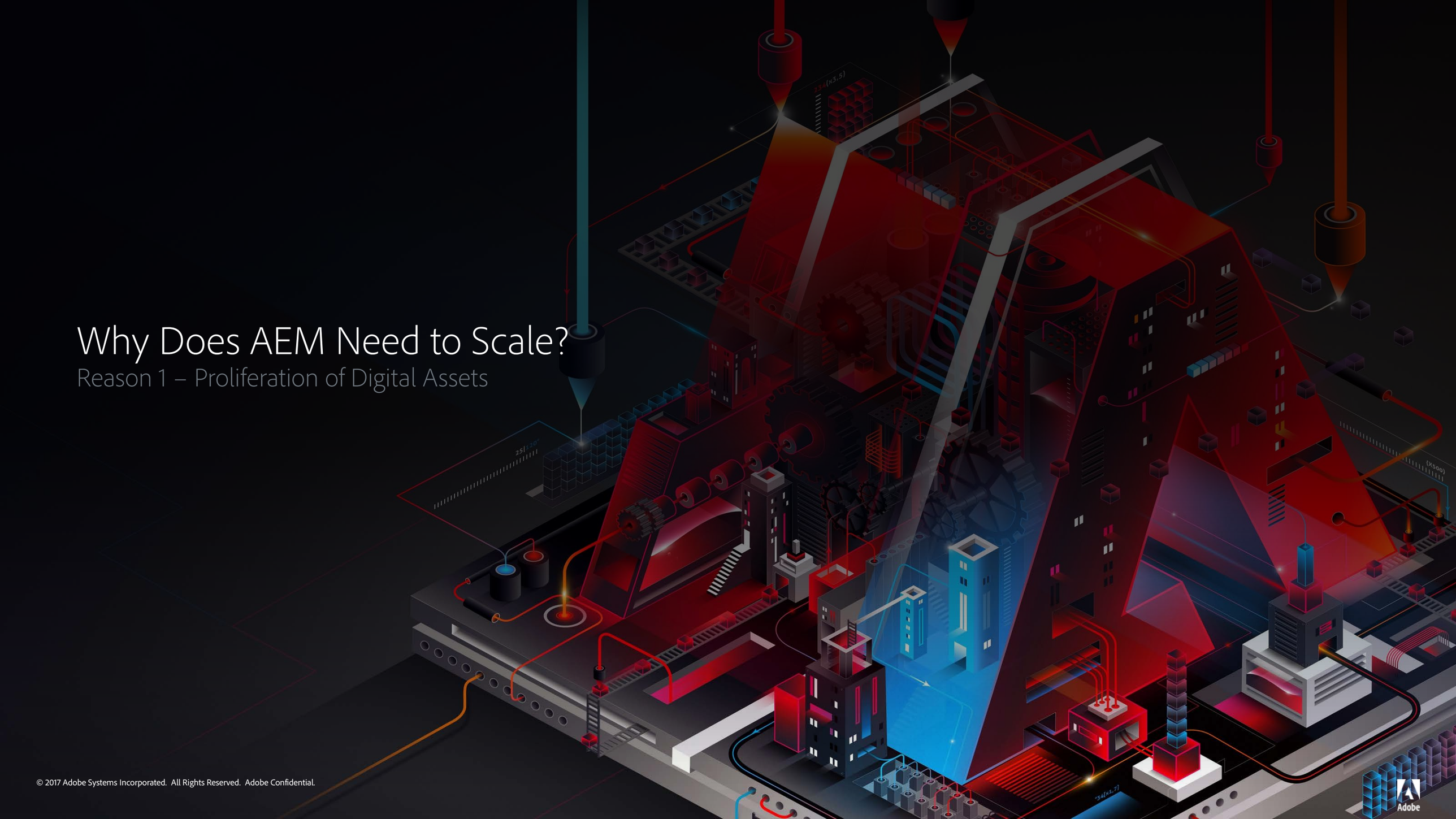
Agenda:

1. Scalability and AEM Authors
    1. Why is it needed?
    2. What is meant by “scalability”?
    3. Where to focus the efforts?
2. What has been done to scale AEM?
3. Q & A

Adobe

Why Does AEM Need to Scale?

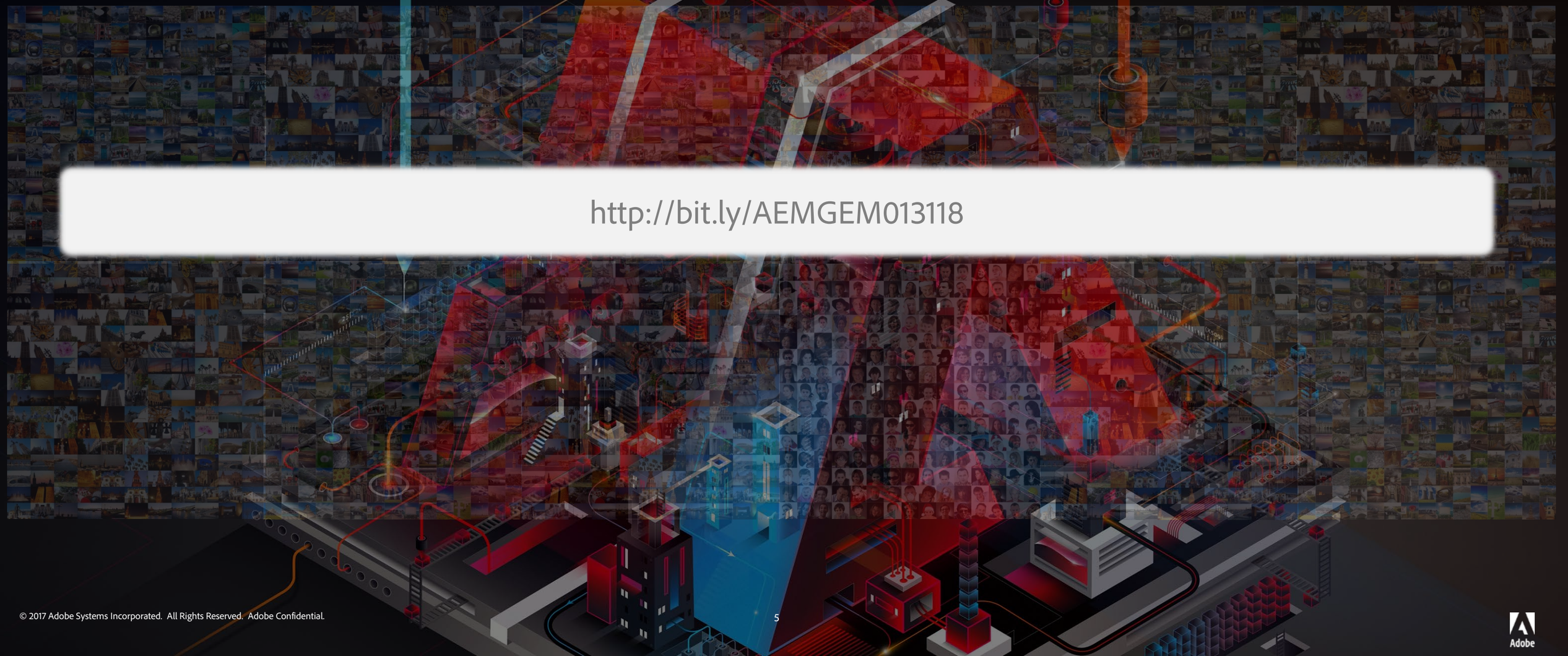
Reason 1 – Proliferation of Digital Assets



The Digital Asset Explosion

Adobe

The Digital Asset Explosion



http://bit.ly/AEMGEM013118

Adobe

Years Ago – Selective in Taking Photos

Years ago, film was expensive and the number of pictures per film pack was low.

As a result, we were much more selective in taking photos.

Adobe

Today – Proliferation of Photos

Today most of us carry a camera around in our pocket, capable of taking and storing thousands of high-quality photos.

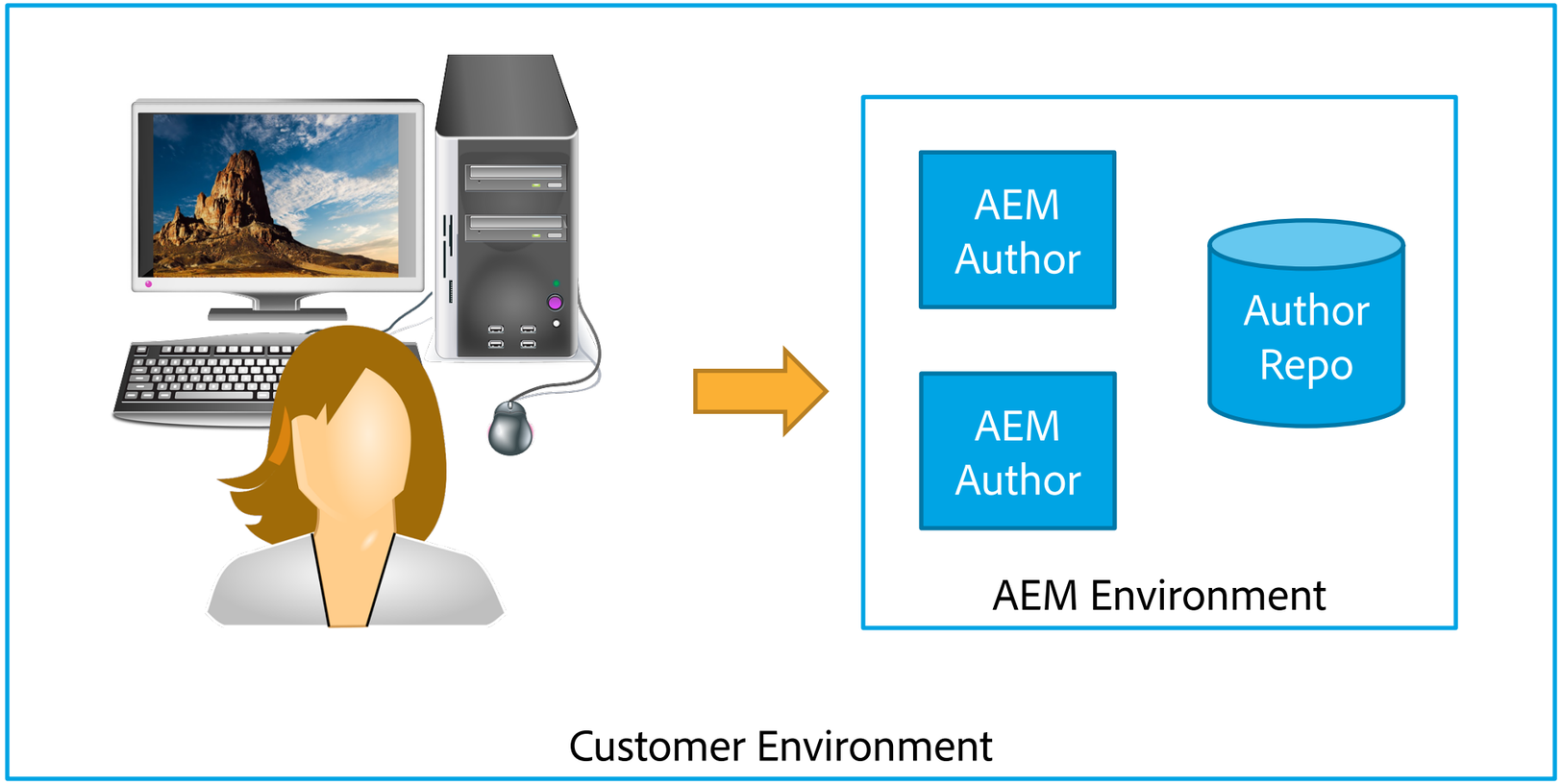
As a result, we are much less selective about the photos we take.

Adobe

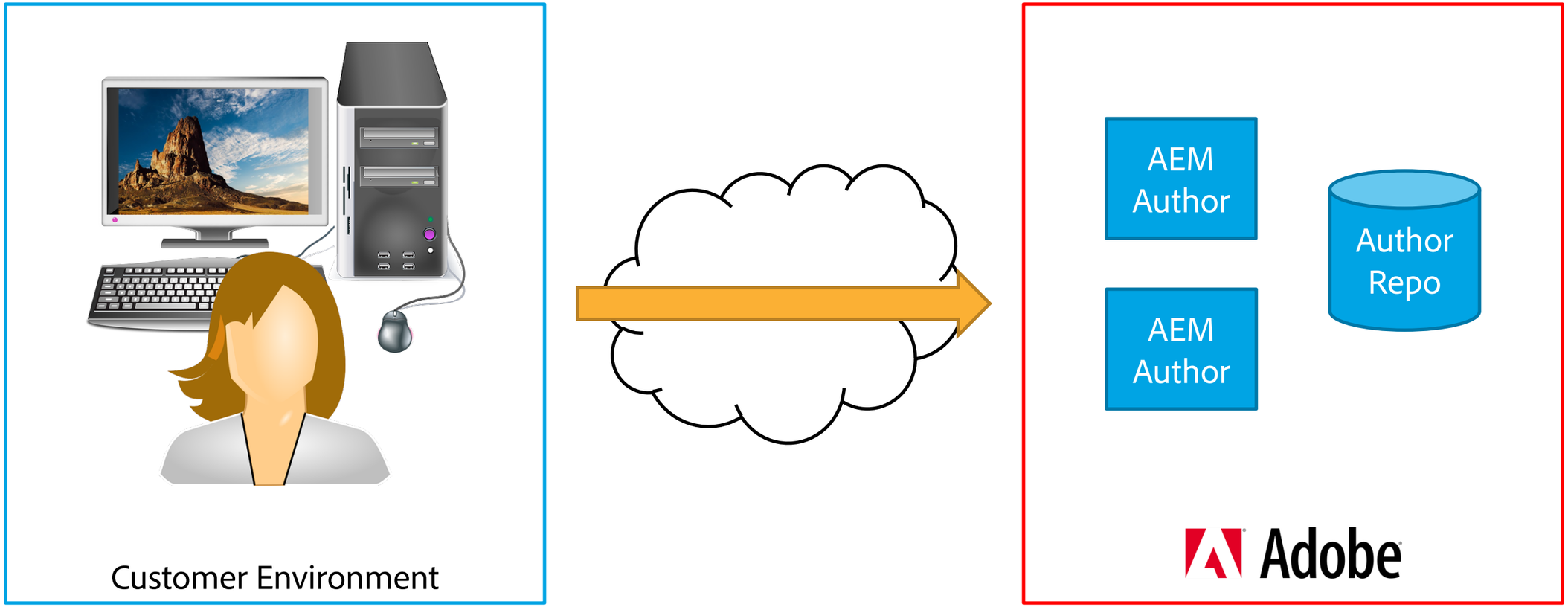
Why Does AEM Need to Scale?

Reason Two – Changing Deployment Models

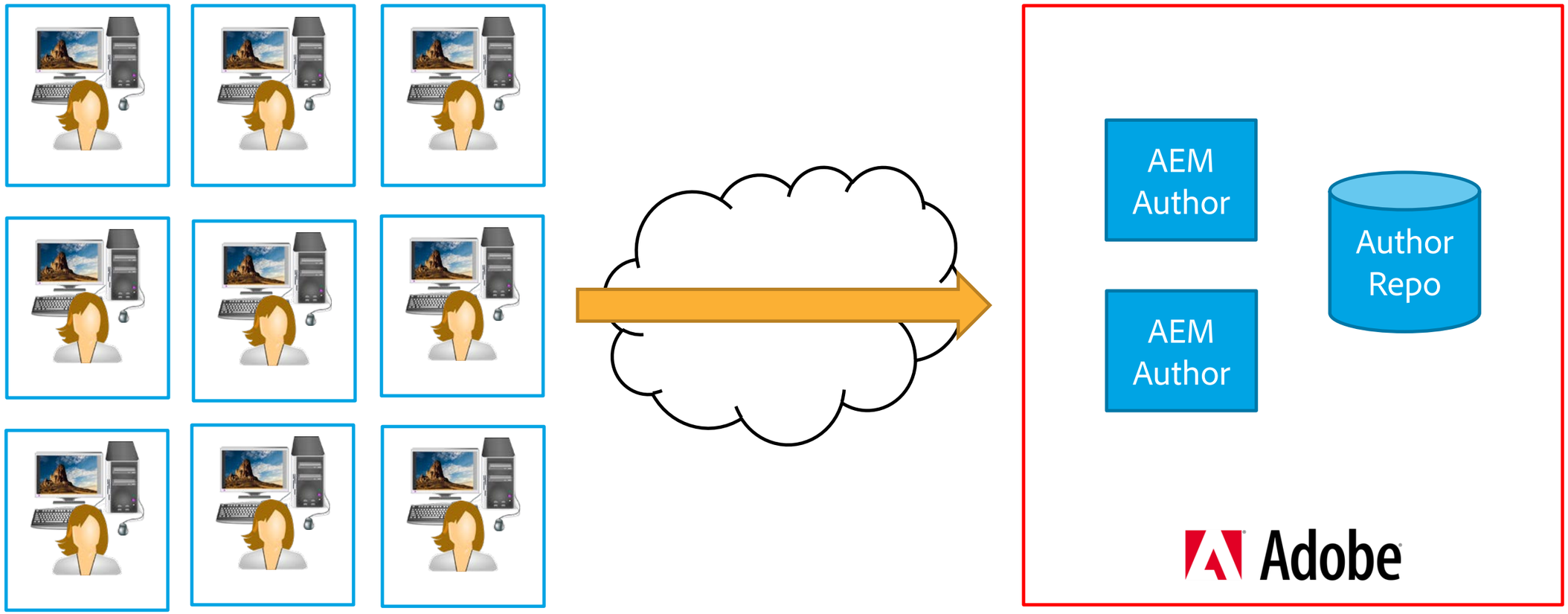
On-Premise AEM to Hosted AEM



On-Premise AEM to Hosted AEM



On-Premise AEM to Hosted AEM



Common Approaches to Application Scalability

Adobe

Common Approaches to Application Scalability

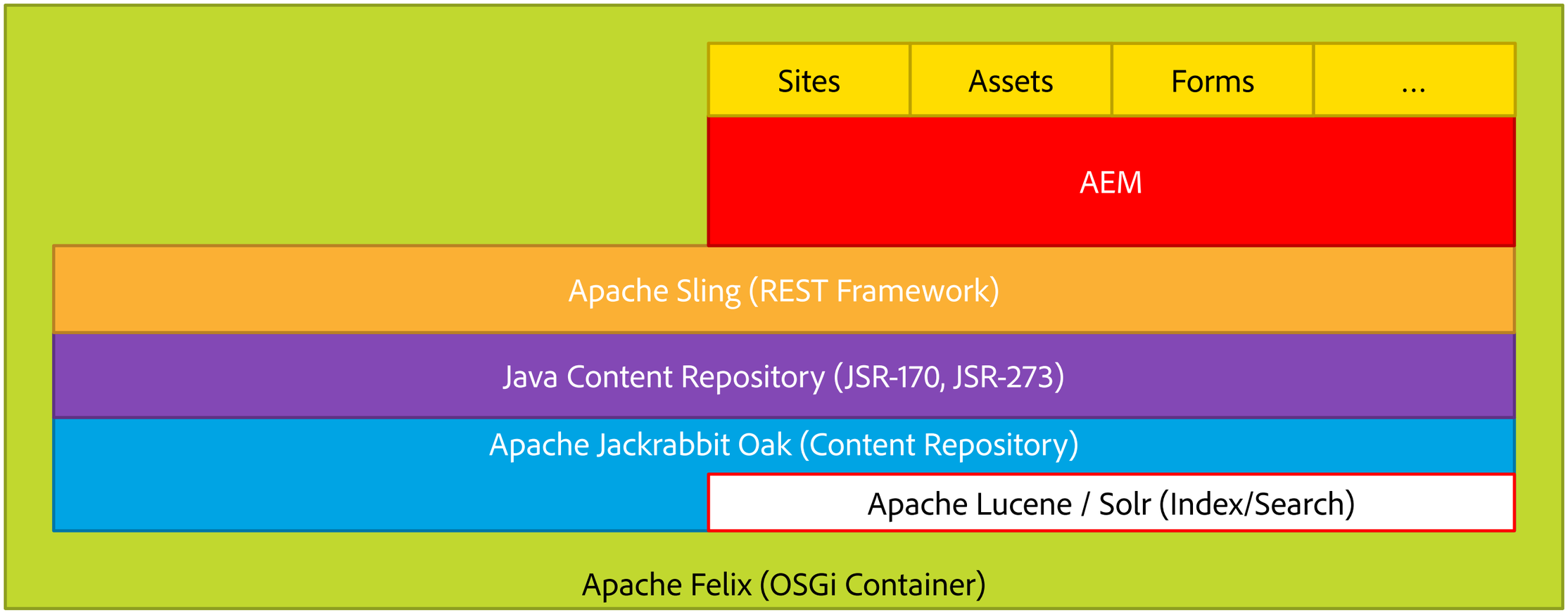
Vertical Scalability

Horizontal Scalability

Data Partitioning

Where Should We Focus Scalability Efforts?

The Architecture of AEM – What Needs to Scale?



Adobe

The Core of AEM – Apache Jackrabbit Oak

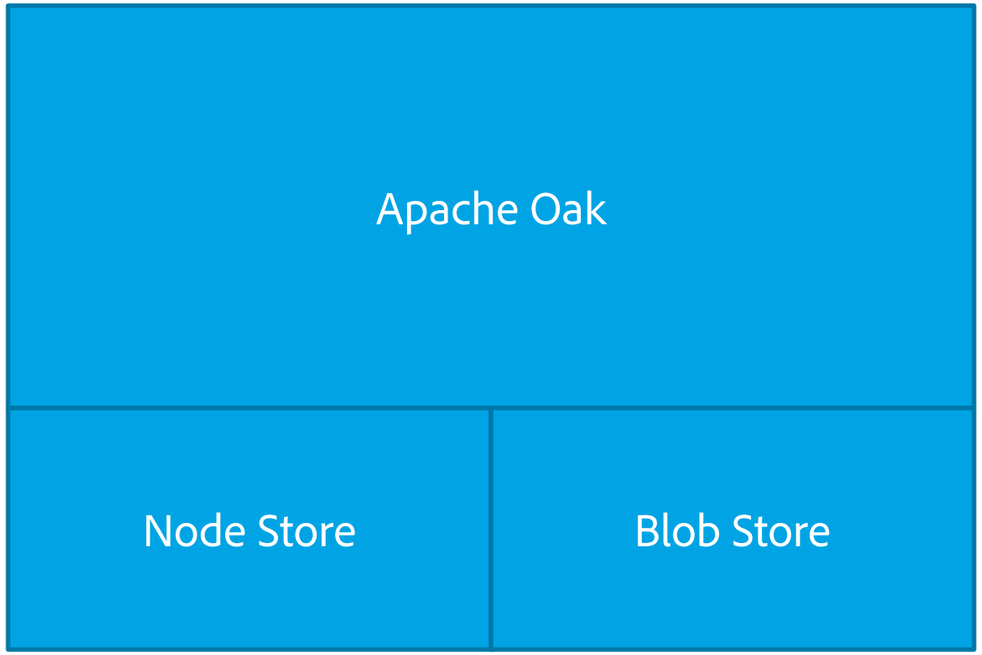
Apache Jackrabbit Oak (Content Repository)

Adobe

Separate Data Storage from Oak

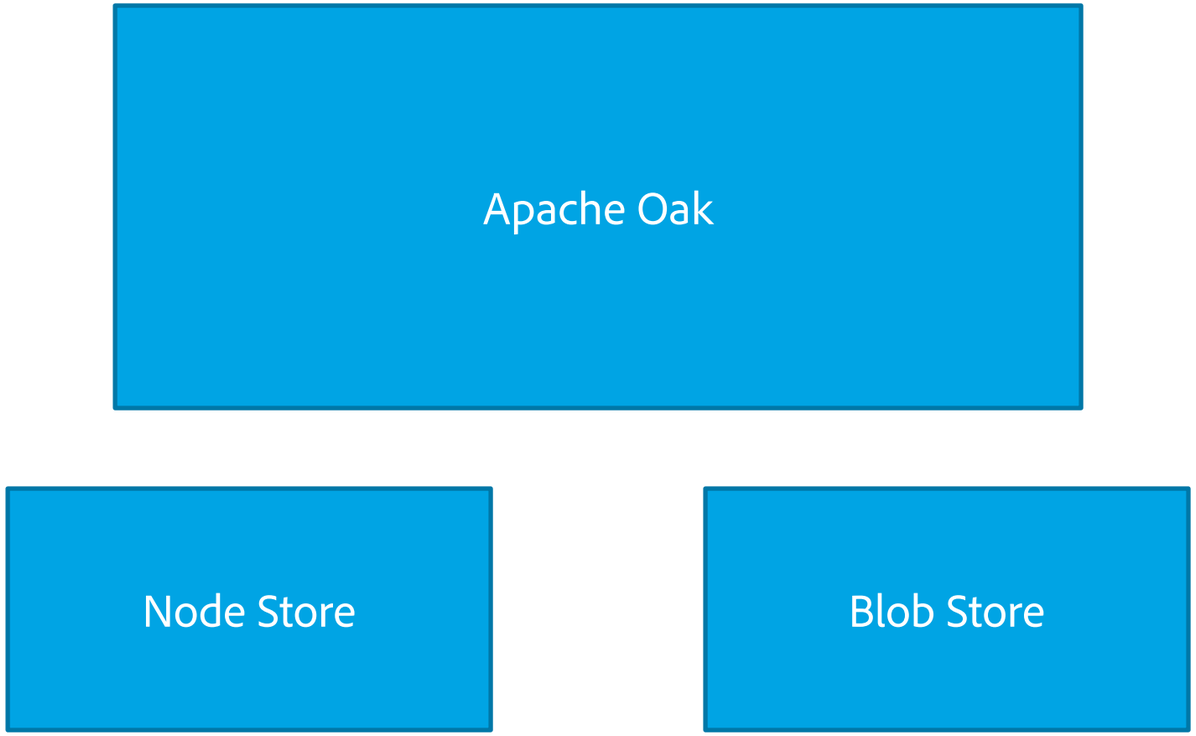
Adobe

Separating Data Storage from Oak



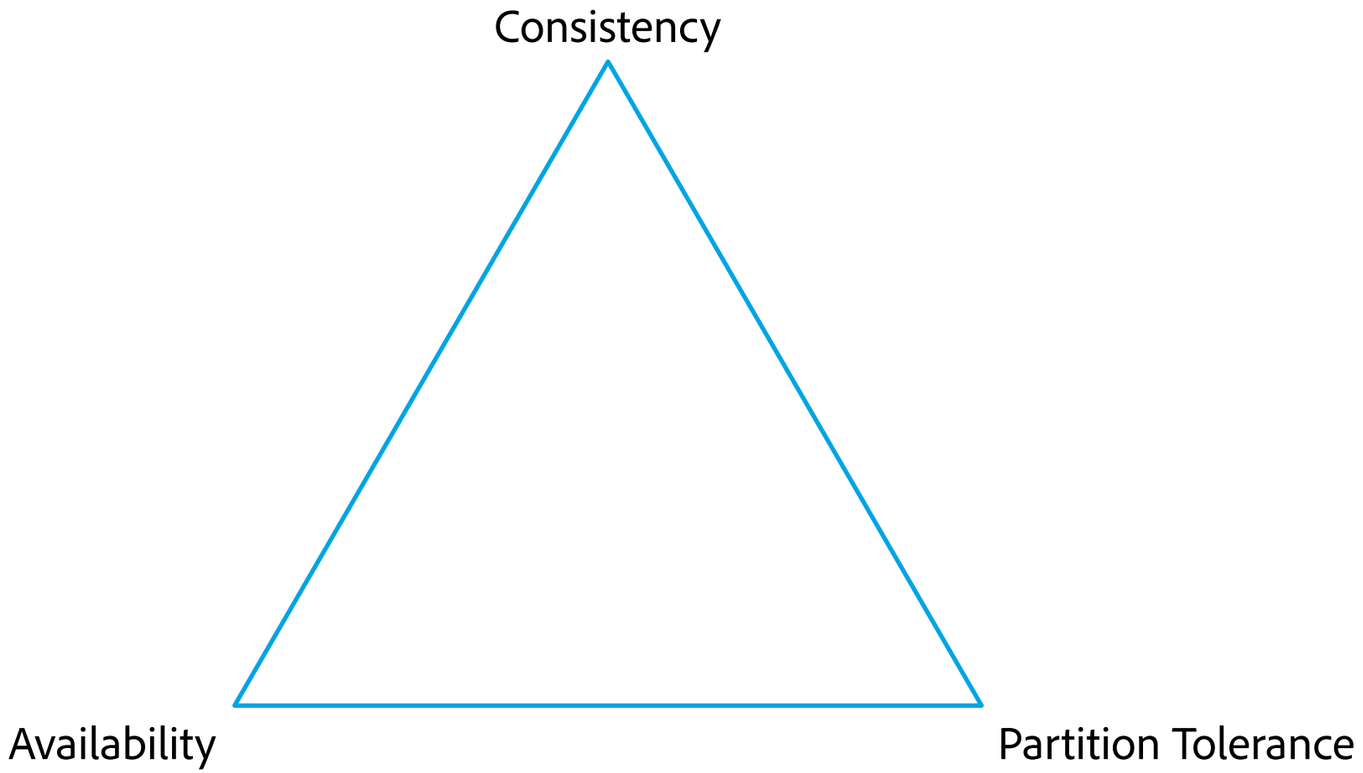
Adobe

Separating Data Storage from Oak

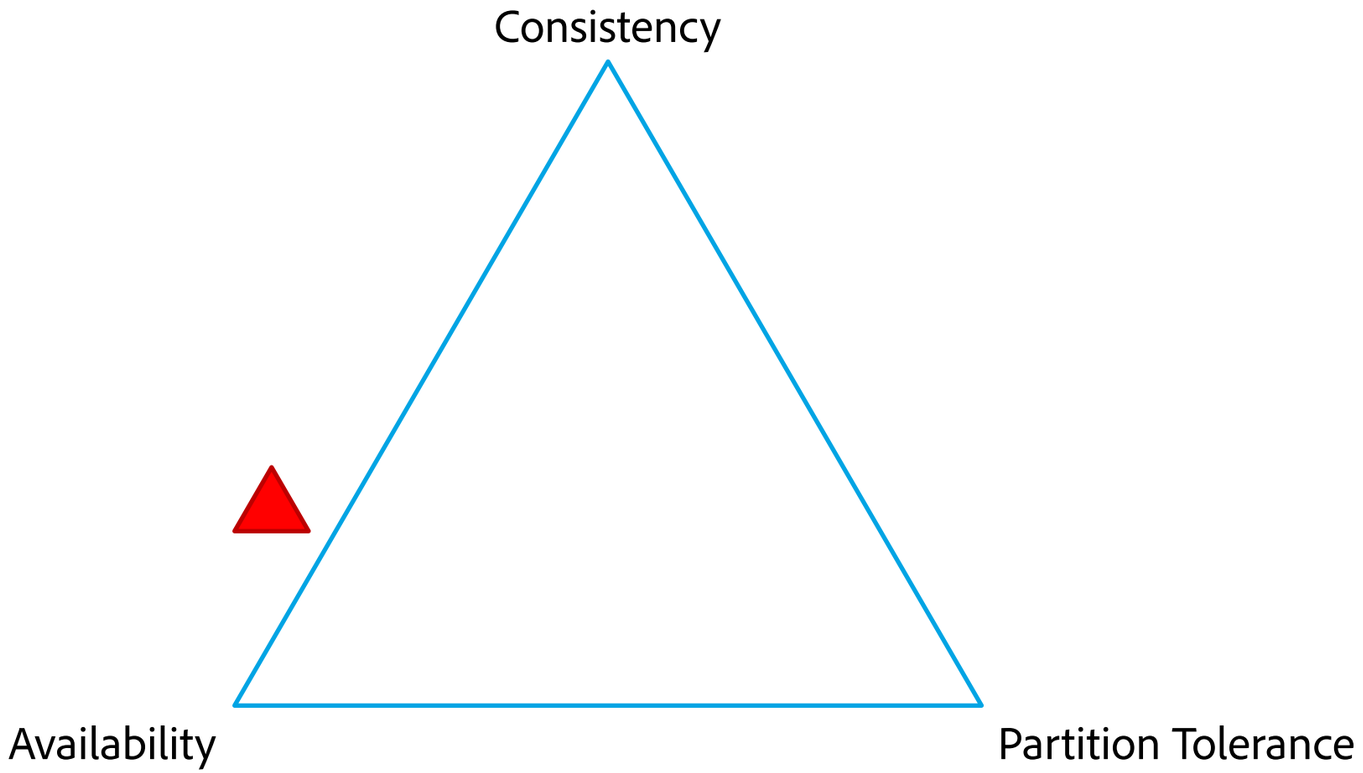


Adobe

CAP Theorem



CAP Theorem and AEM



Scaling the Blob Store

Scalable blob storage is a solved problem.

Adobe

Scalable database storage is a solved problem.

Adobe

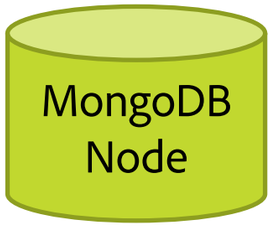
Scalable database storage is a solved problem…?

What is the consistency model of MongoDB?

Adobe

A Hypothetical MongoDB Deployment

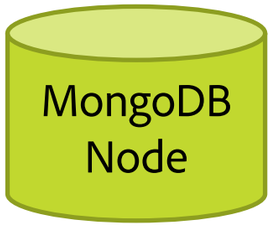
write=0

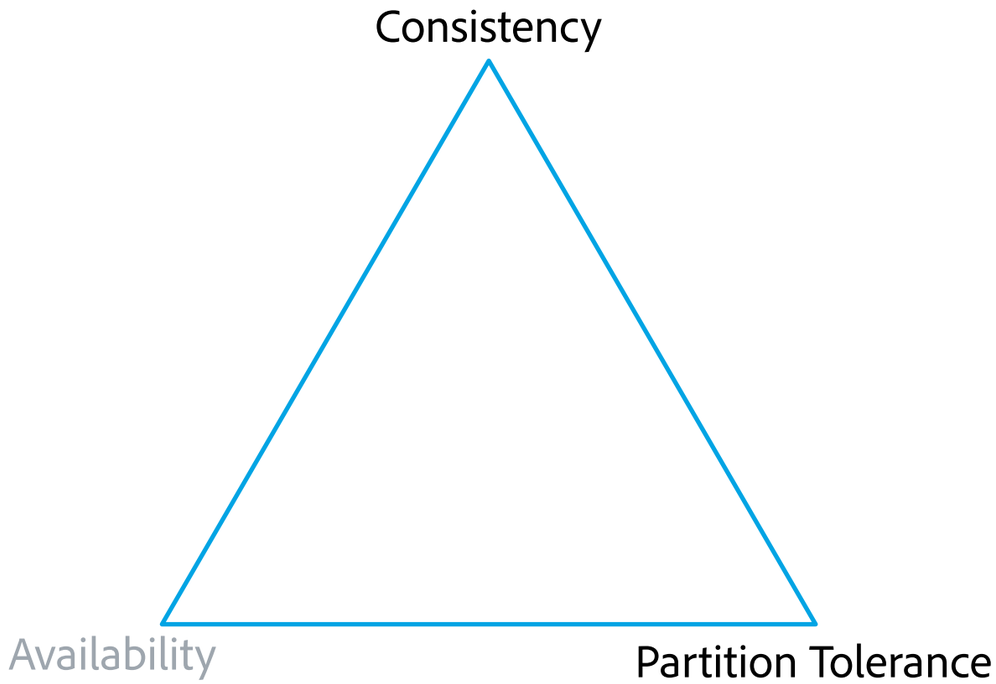


Adobe

A Hypothetical MongoDB Deployment

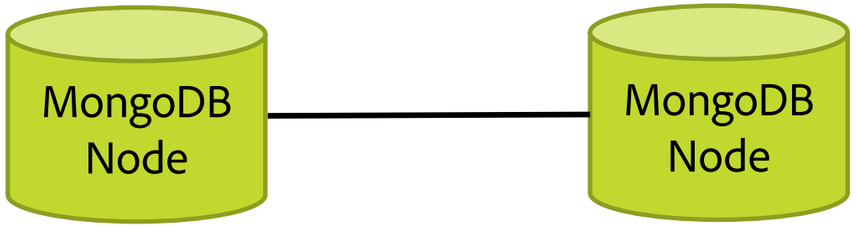
~~write=0~~
write=1





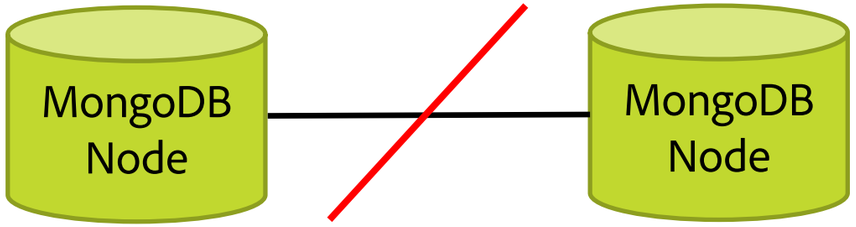
A Hypothetical MongoDB Deployment

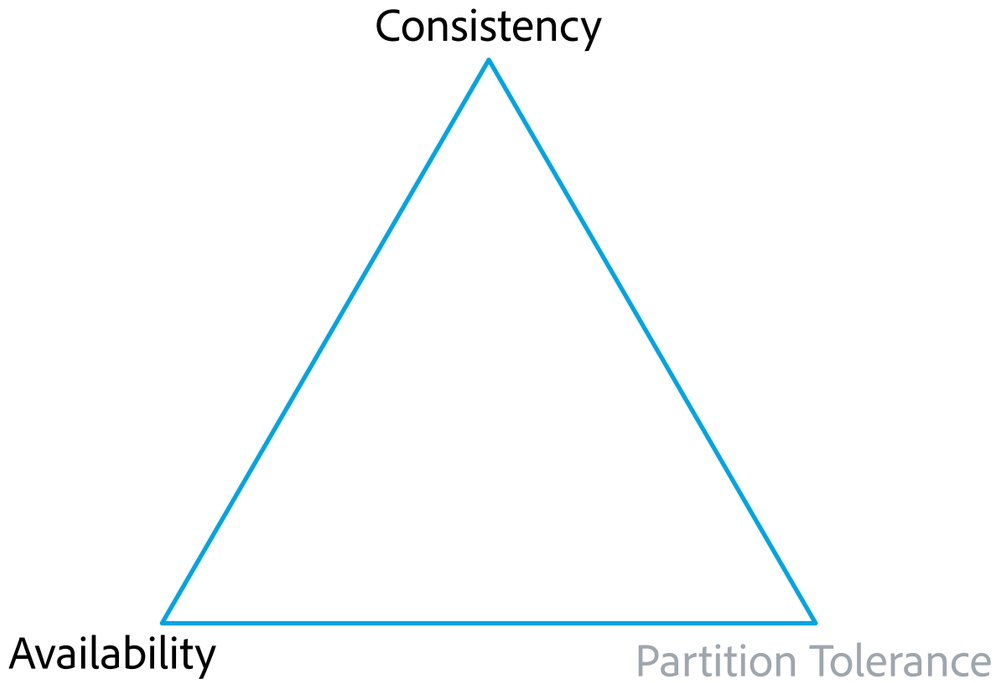
~~write=0~~
write=1



A Hypothetical MongoDB Deployment

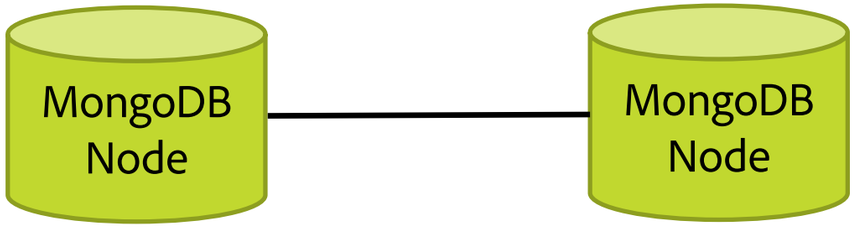
~~write=0~~
write=1





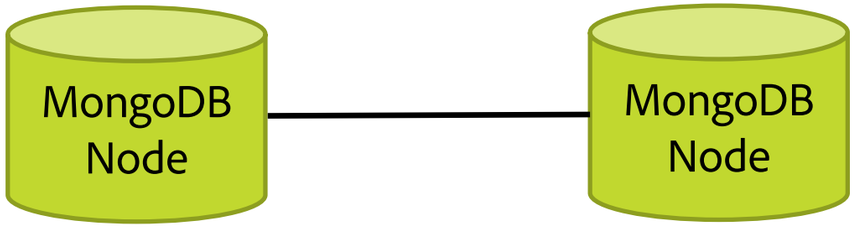
A Hypothetical MongoDB Deployment

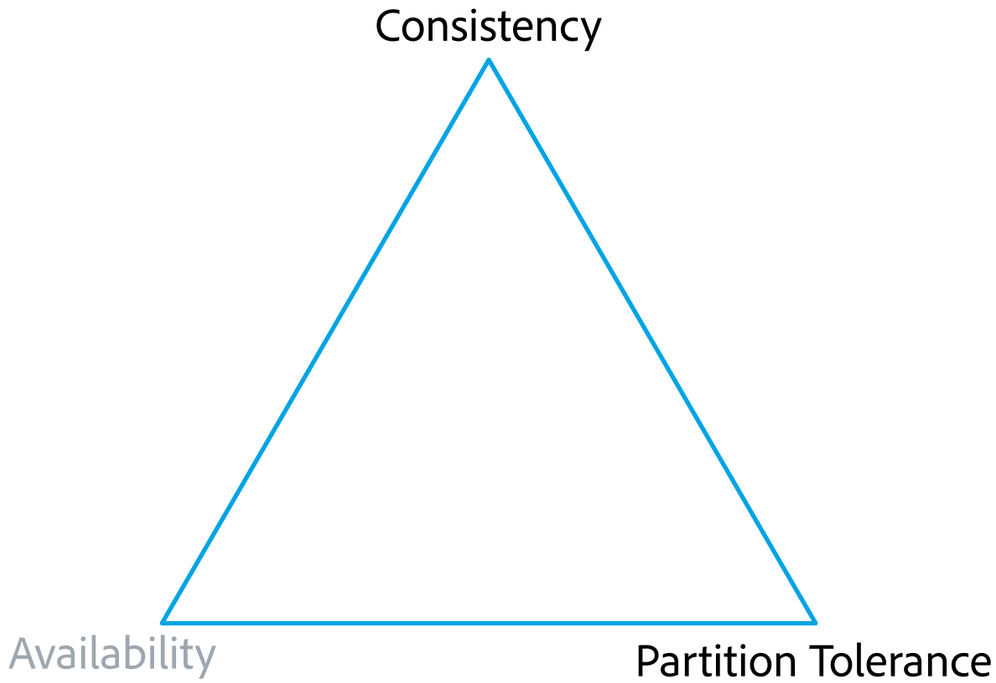
~~write=0~~
~~write=1~~
write=2



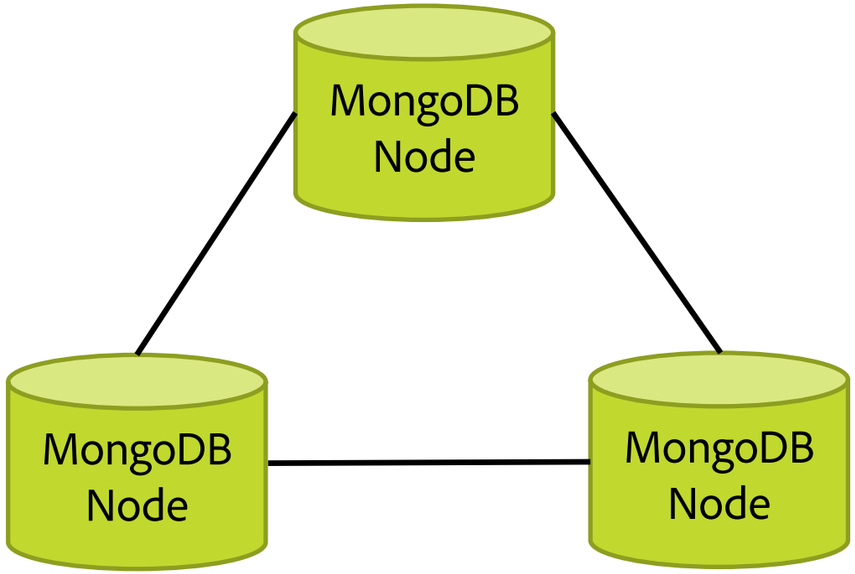
A Hypothetical MongoDB Deployment

~~write=0~~
~~write=1~~
write=2

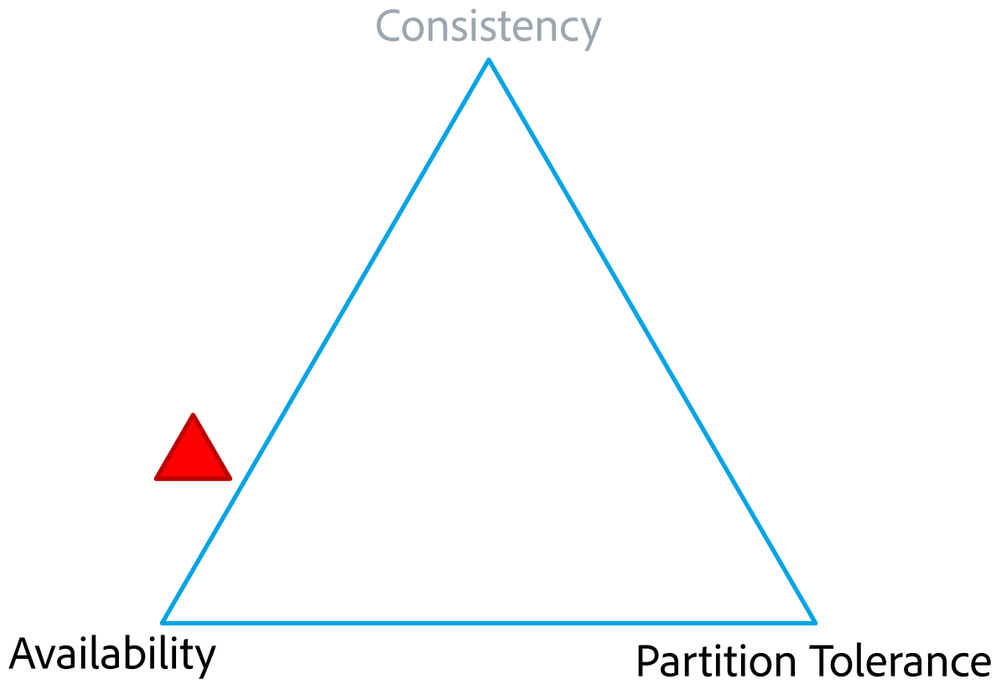




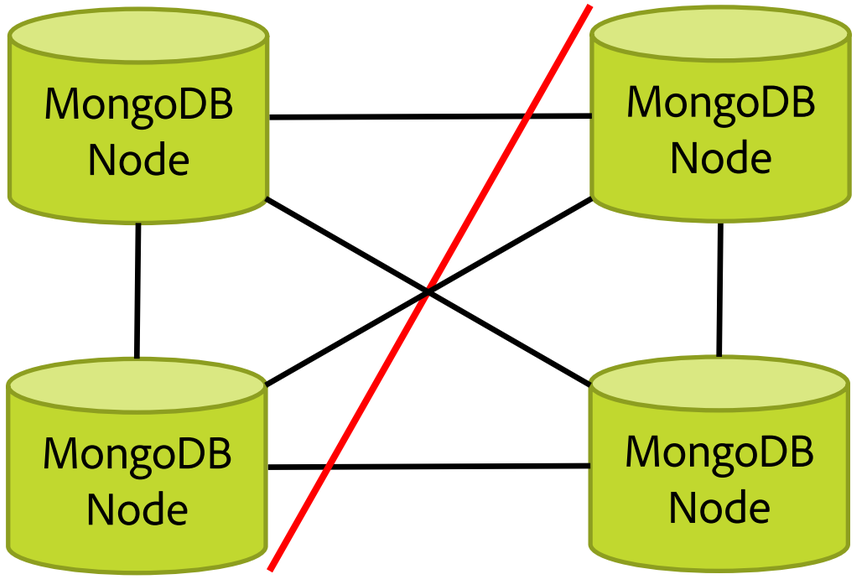
A Hypothetical MongoDB Deployment



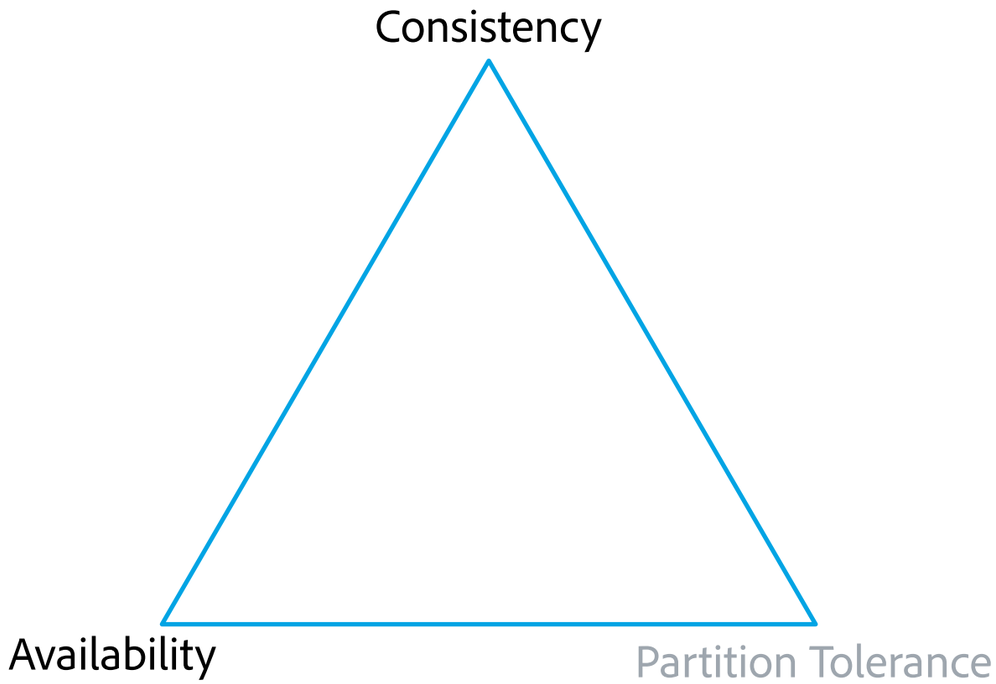
~~write=0~~
~~write=1~~
write=2



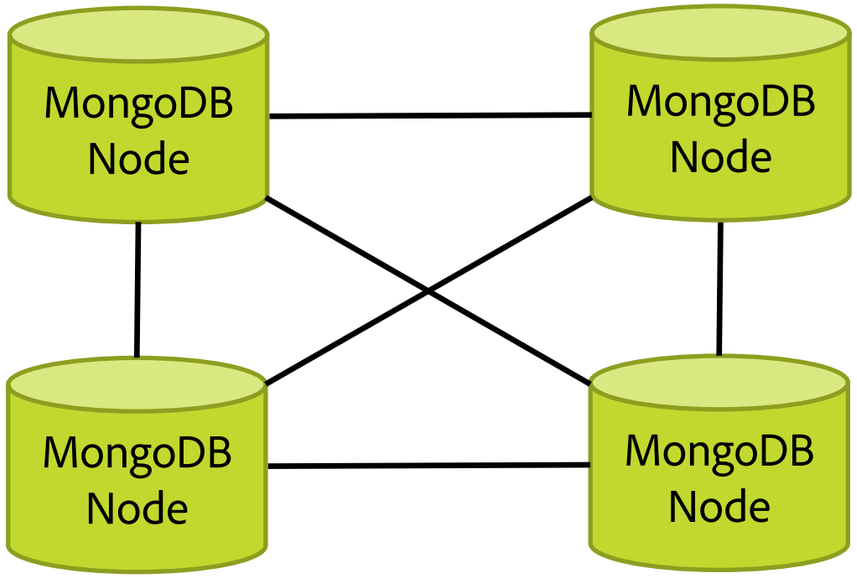
A Hypothetical MongoDB Deployment



~~write=0~~
~~write=1~~
write=2

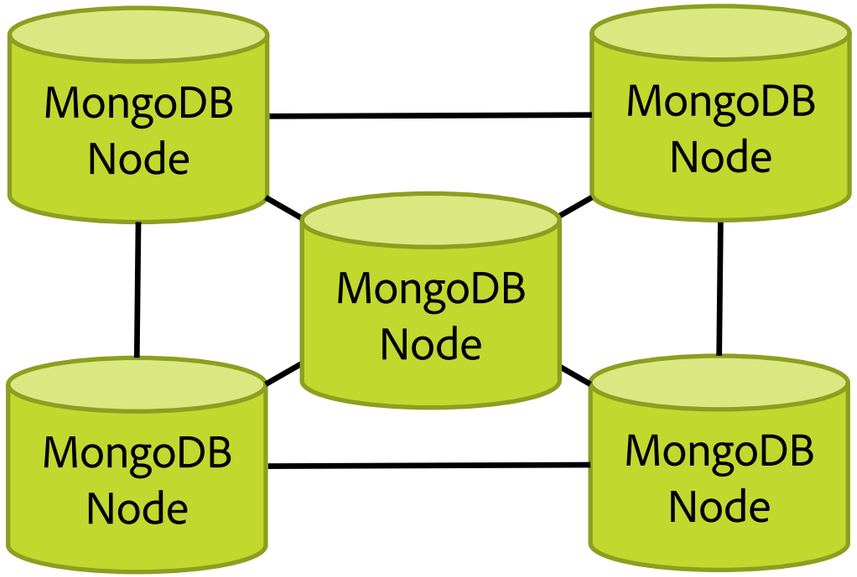


A Hypothetical MongoDB Deployment



~~write=0~~
~~write=1~~
~~write=2~~
write=3

A Hypothetical MongoDB Deployment



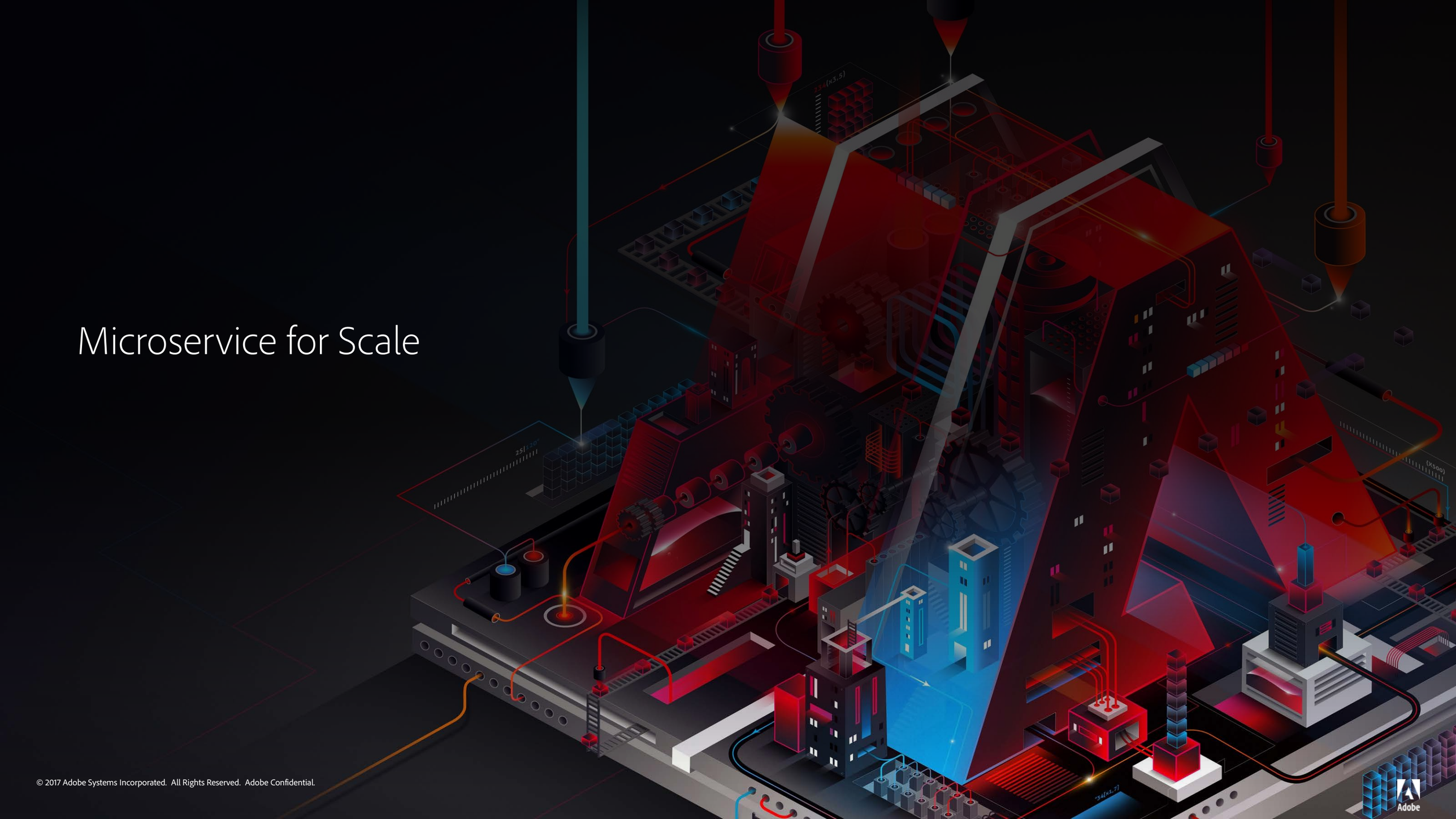
~~write=0~~
~~write=1~~
~~write=2~~
write=3

Takeaway

Math and physics says there are limits to how far you can reasonably scale a database.

Adobe

Microservice for Scale

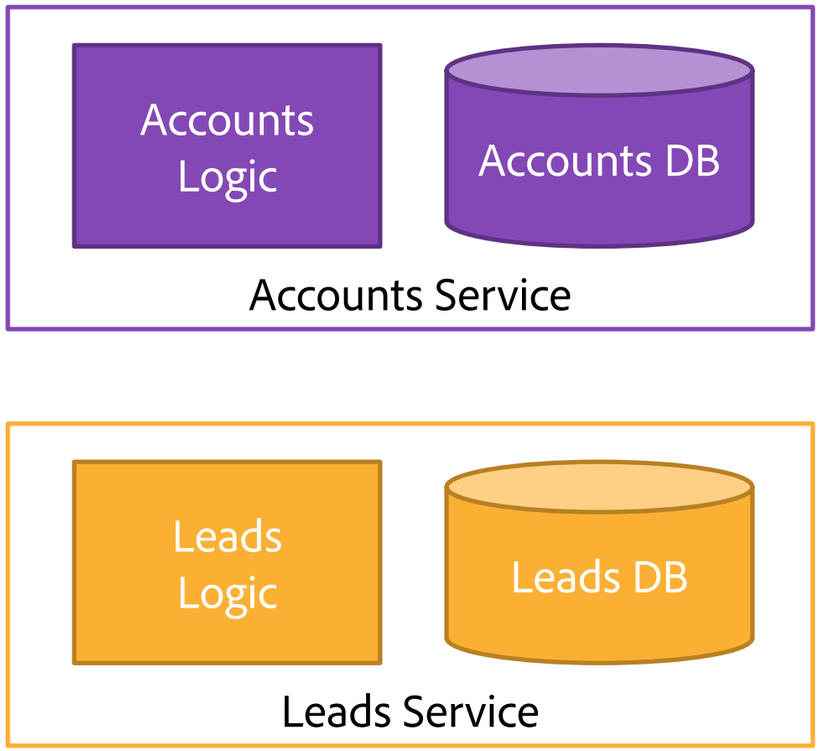


Microservices!!!

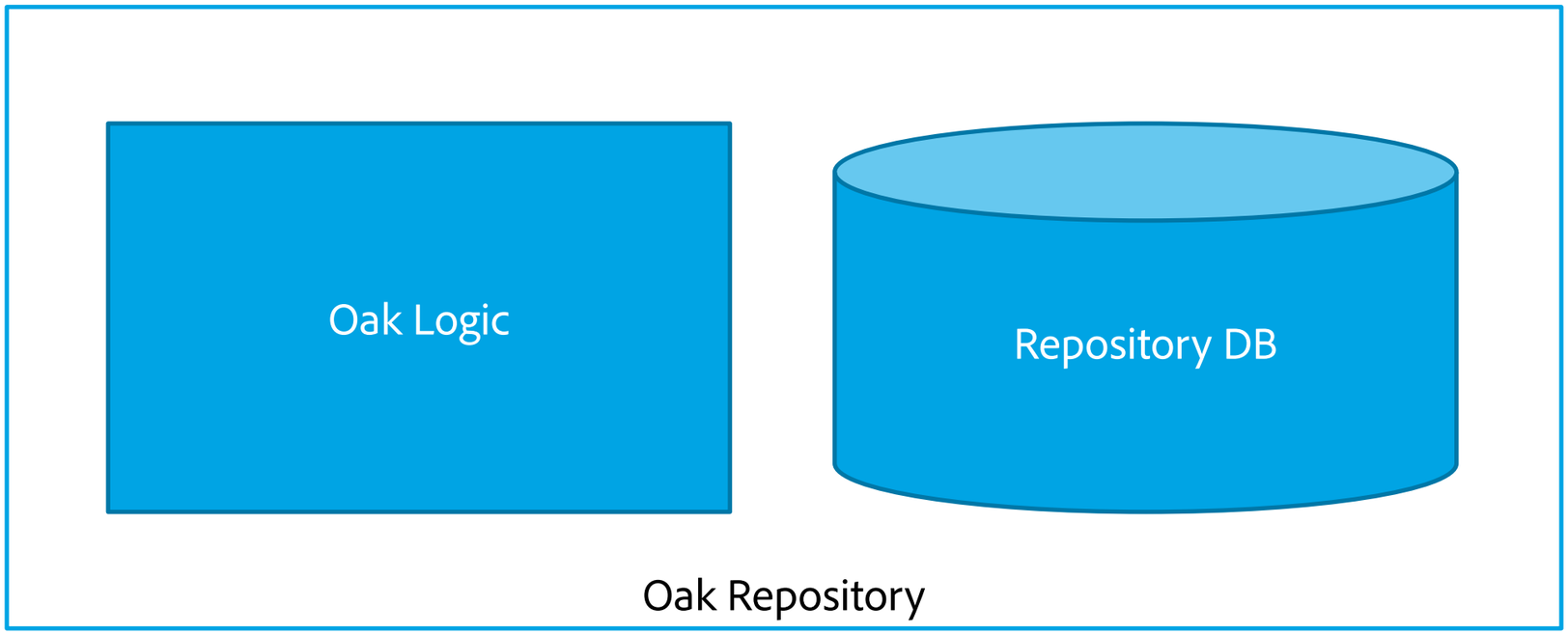
We should microservice Oak.

What is a Microservice?

- Small, autonomous services that work together[1]
- Single Responsibility Principle
- Independently deployable, manageable, and scalable
- Loose couplings
  - No shared data



Splitting Oak into Microservices



- What are Oak's responsibilities?
- How do you split the data for Oak?

Microservices???

Should we microservice Oak?

Takeaway

Microservices are inseparable from their data.

Scalability Challenge Number One

You cannot split a data store into true microservices.

Splitting AEM by Function (Not True Microservices)

Adobe

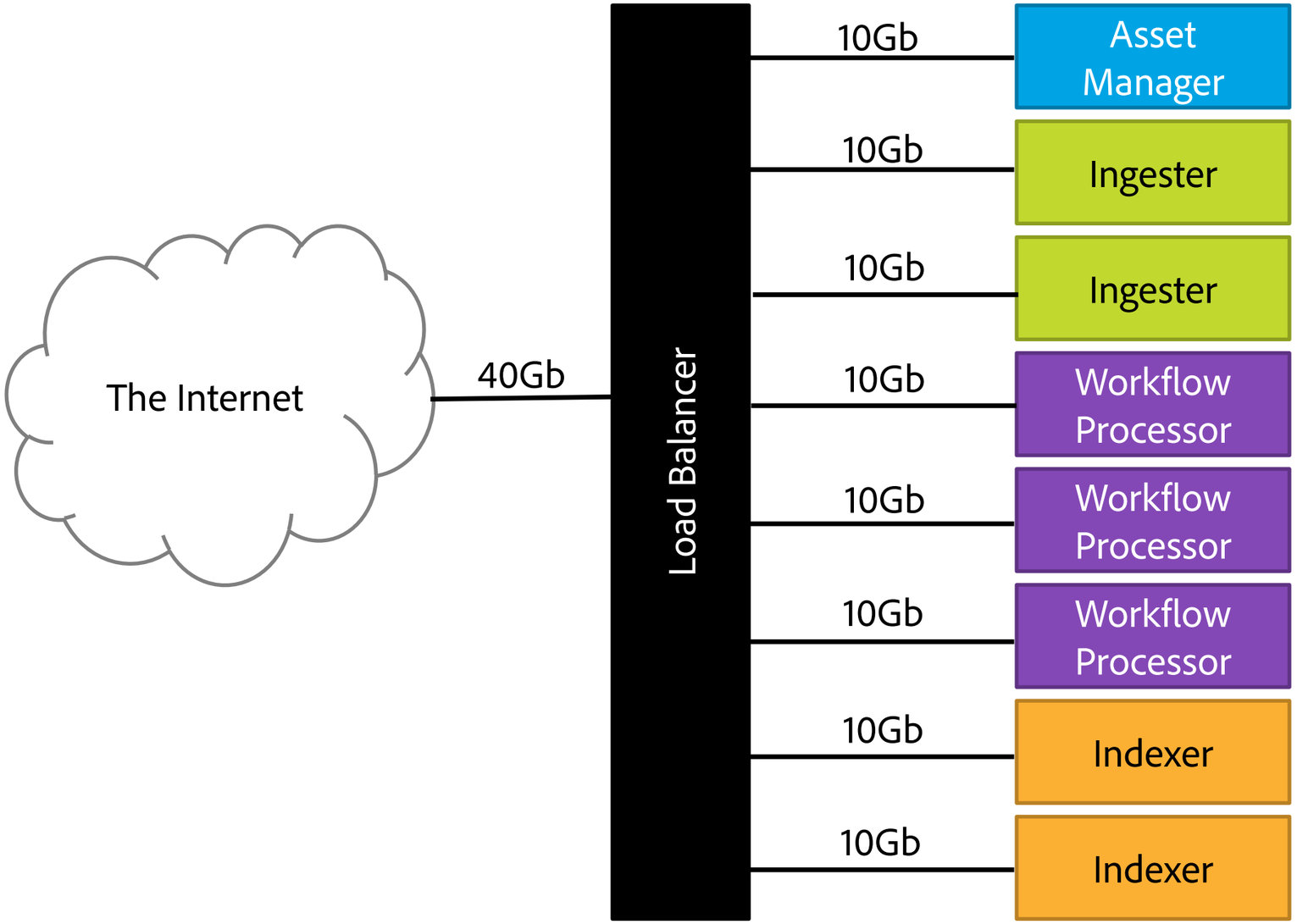
Splitting out AEM Functionality

Adobe

Splitting out AEM Functionality

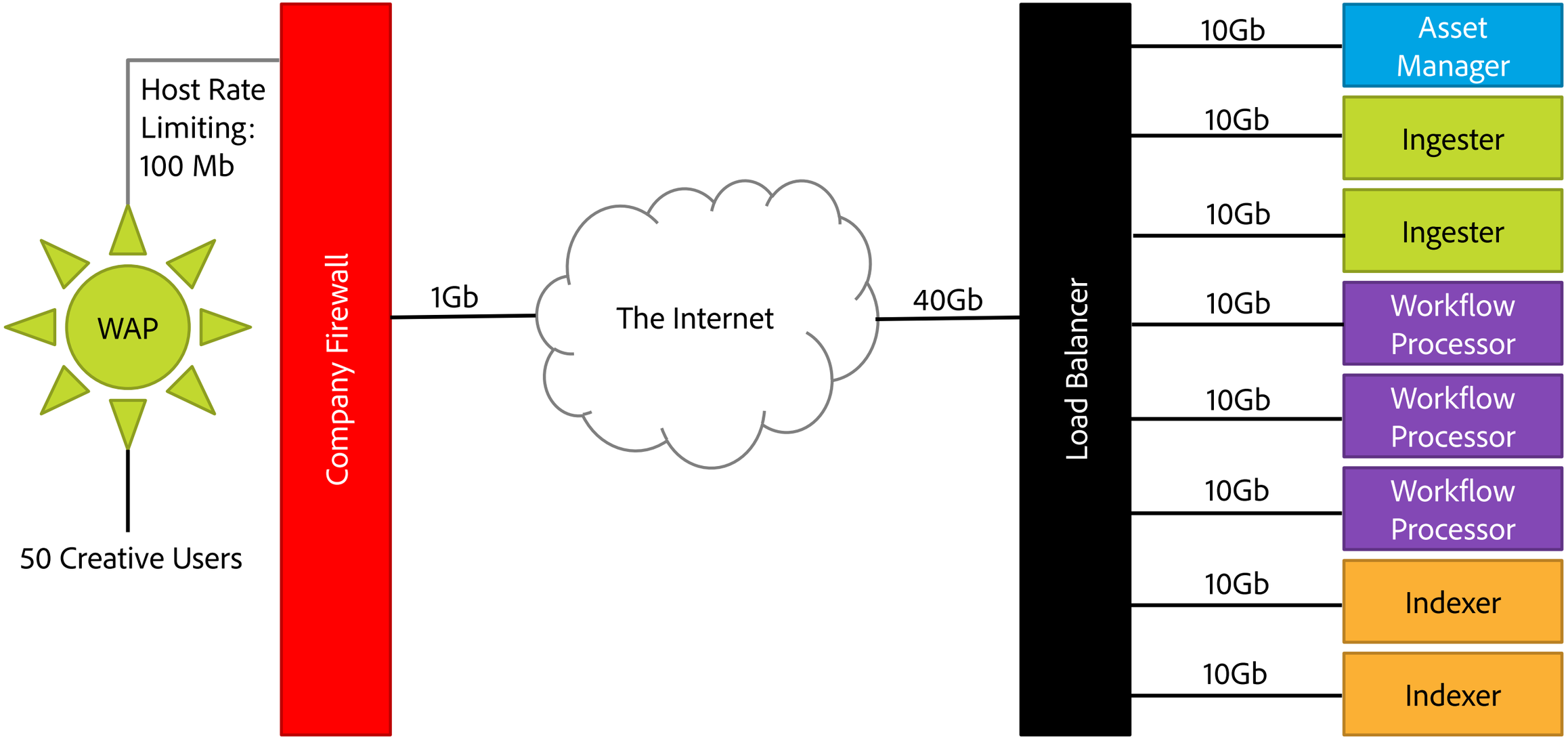
Adobe

Problem Solved



Adobe

Problem Solved…???



Zero Problems Solved

Adobe

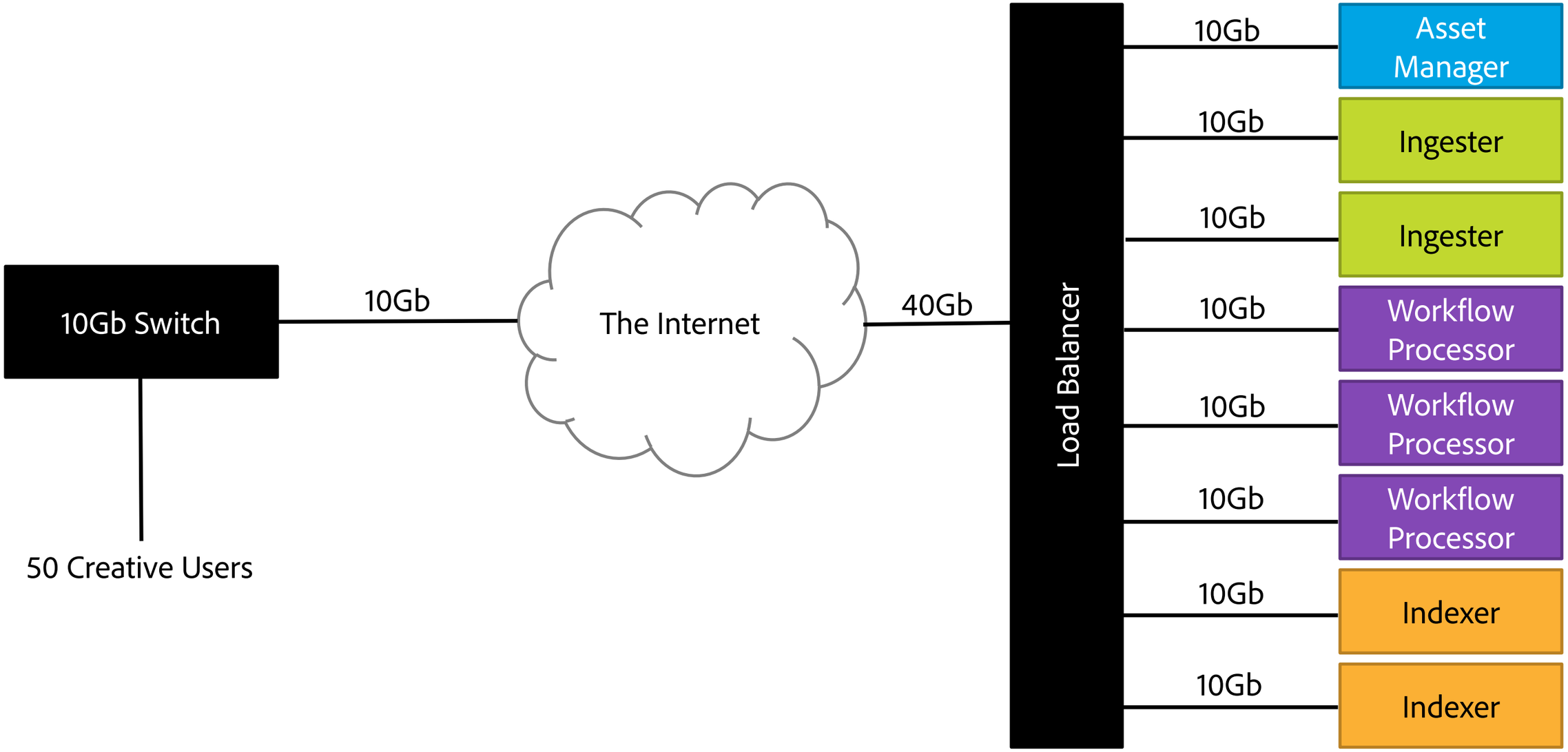
Scalability Challenge Number Two

LAN infrastructure may limit the ability to scale effectively.

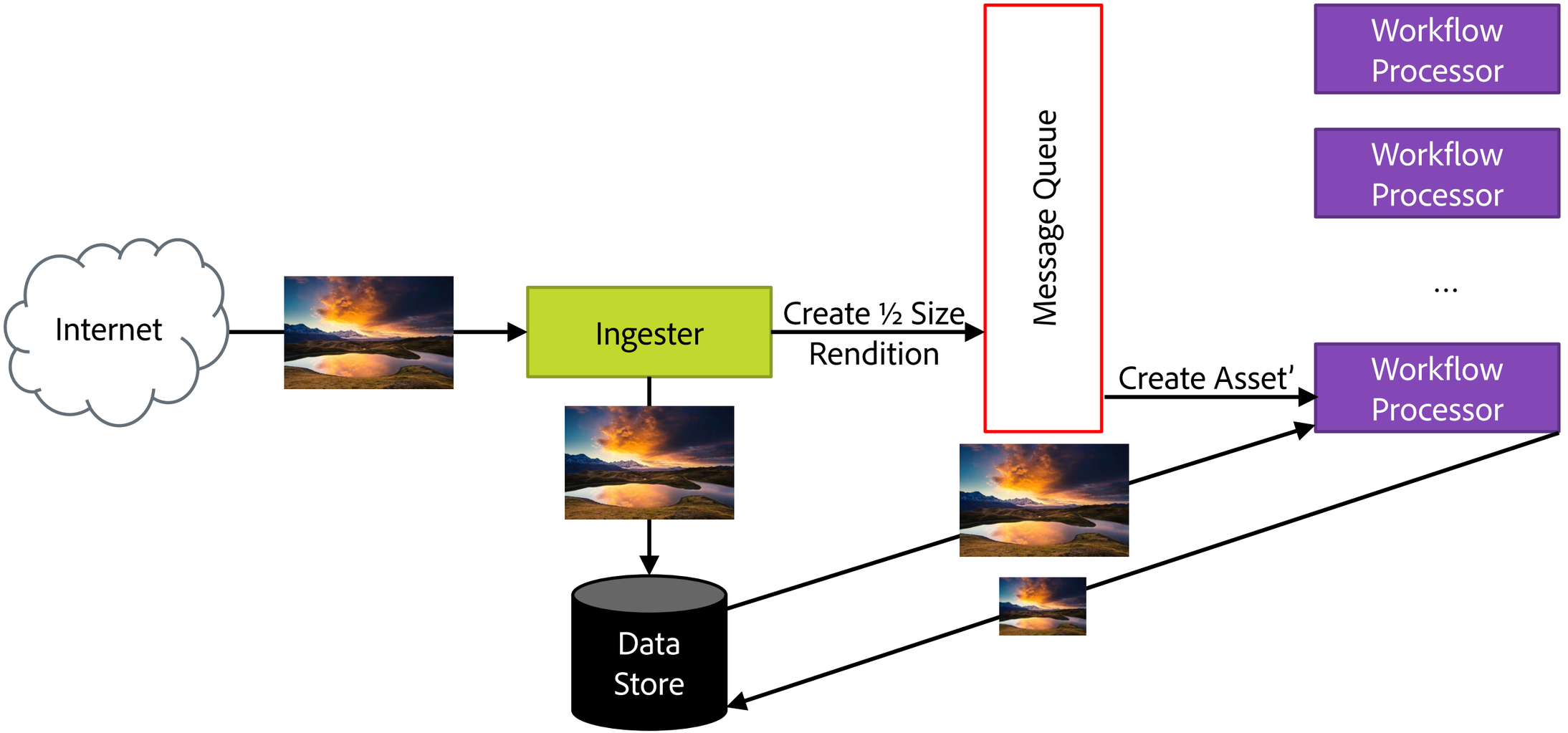
Takeaway

Be clear about the benefits you hope to gain when refactoring for scalability.

Problem Solved…???



Scaling the Ingestion Process

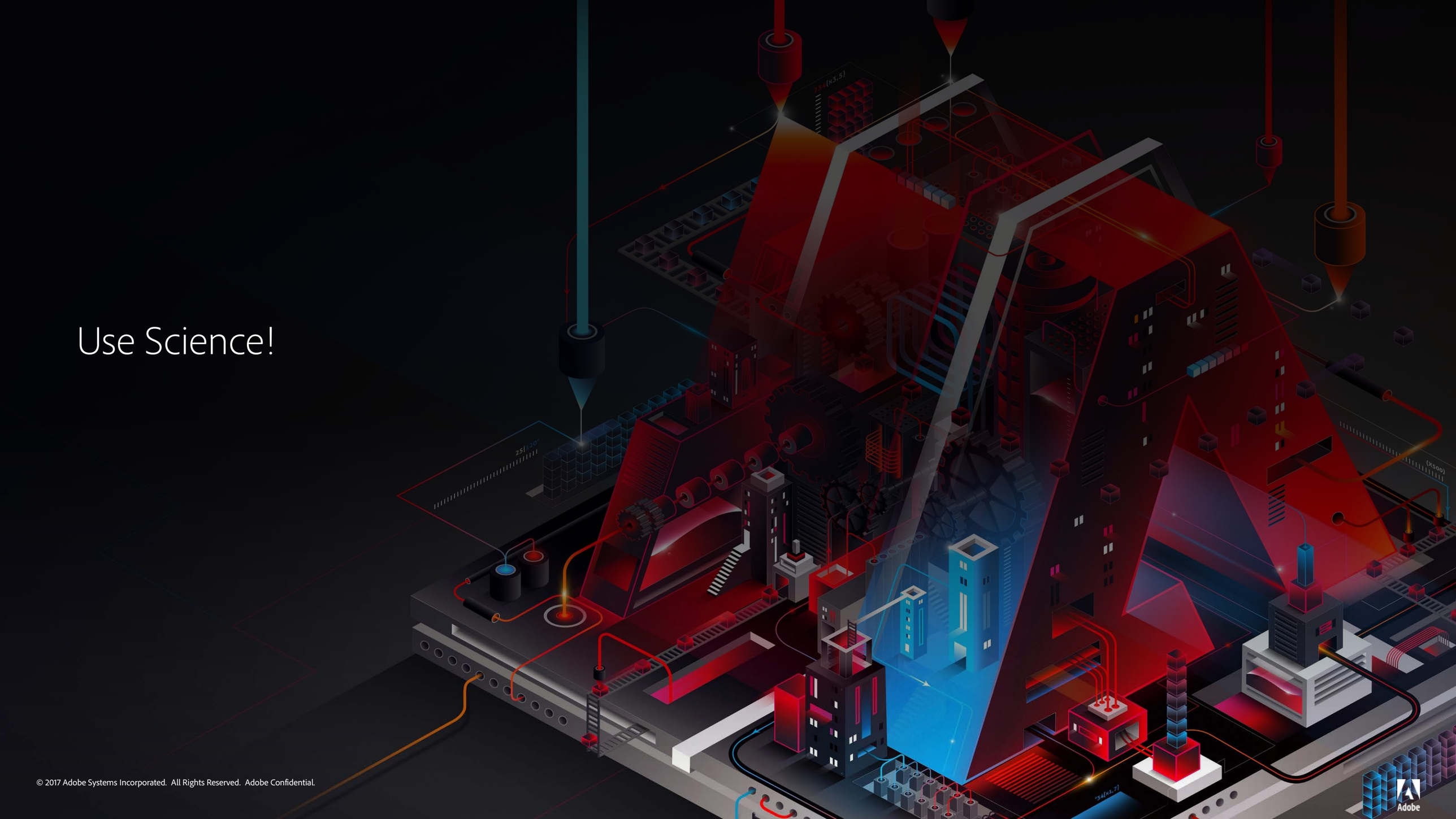


Is it worth the expense?

1. Research message queueing systems to determine best fit
2. Adding a message queueing system
3. Modify code to use a queue
4. Modify code to us an asynchronous model
5. Split codebase into distinct services
6. Run services on separate infrastructure
7. Testing

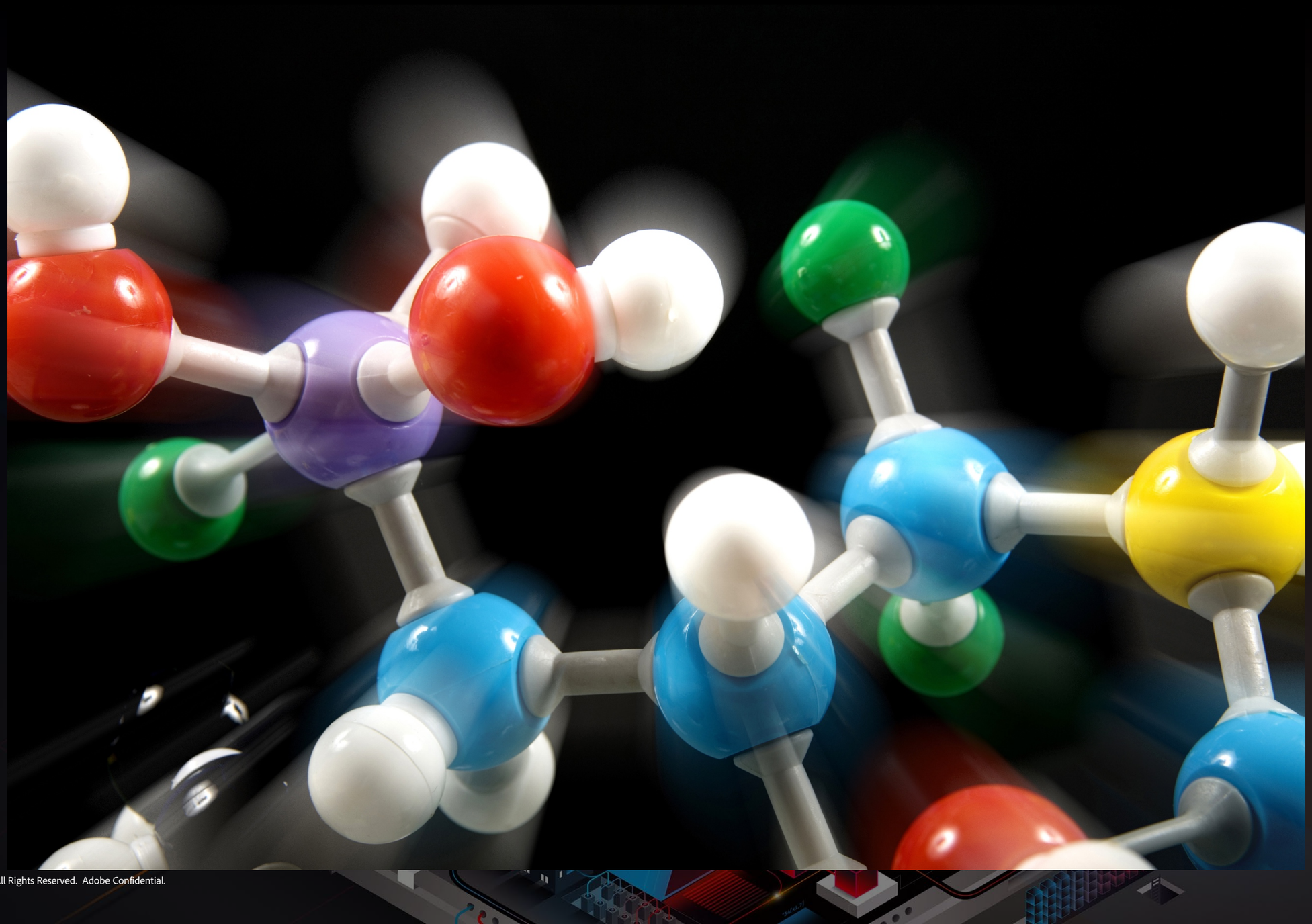
Adobe

Use Science!



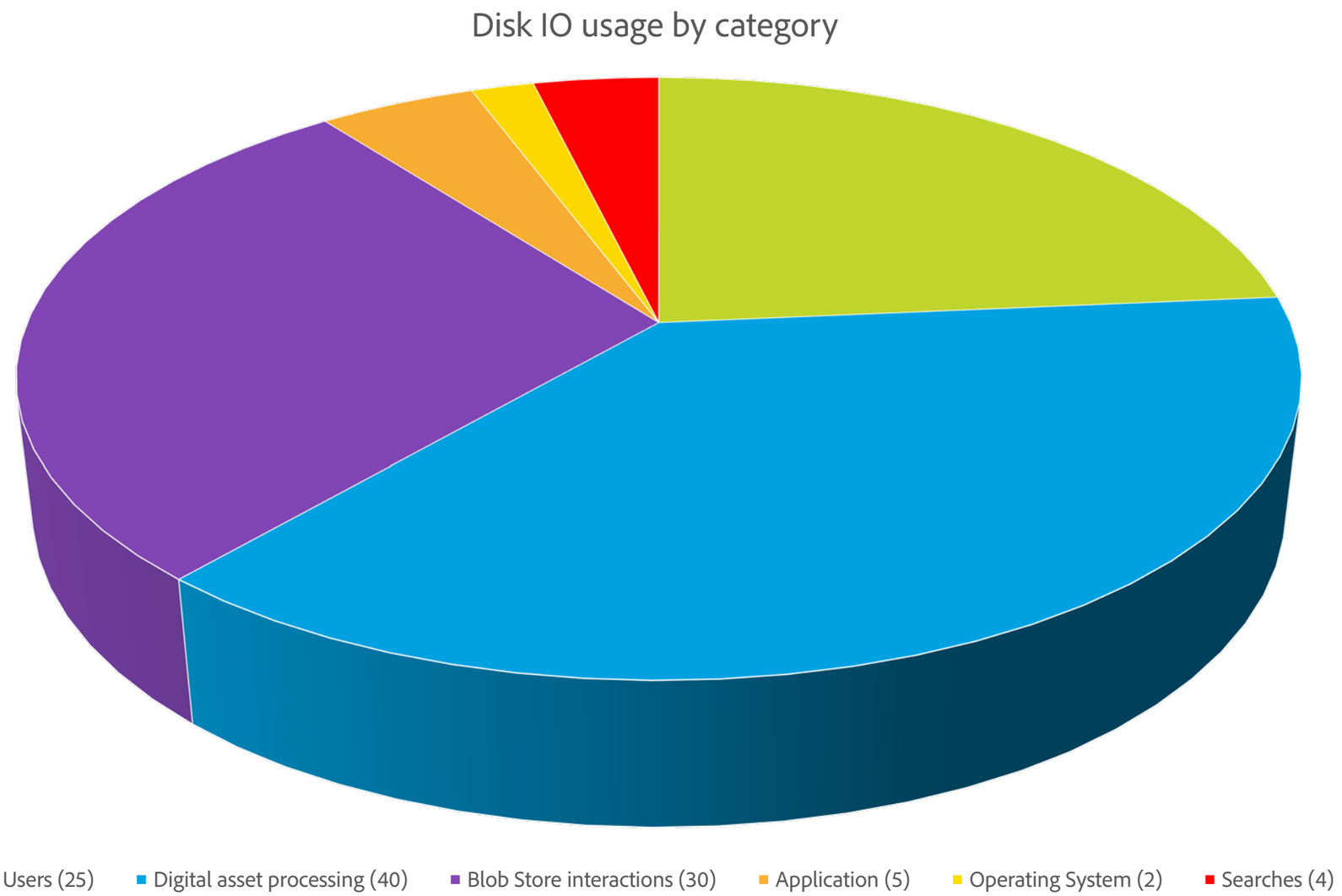
Question

What limits AEM performance?
Is AEM CPU bound or I/O bound?

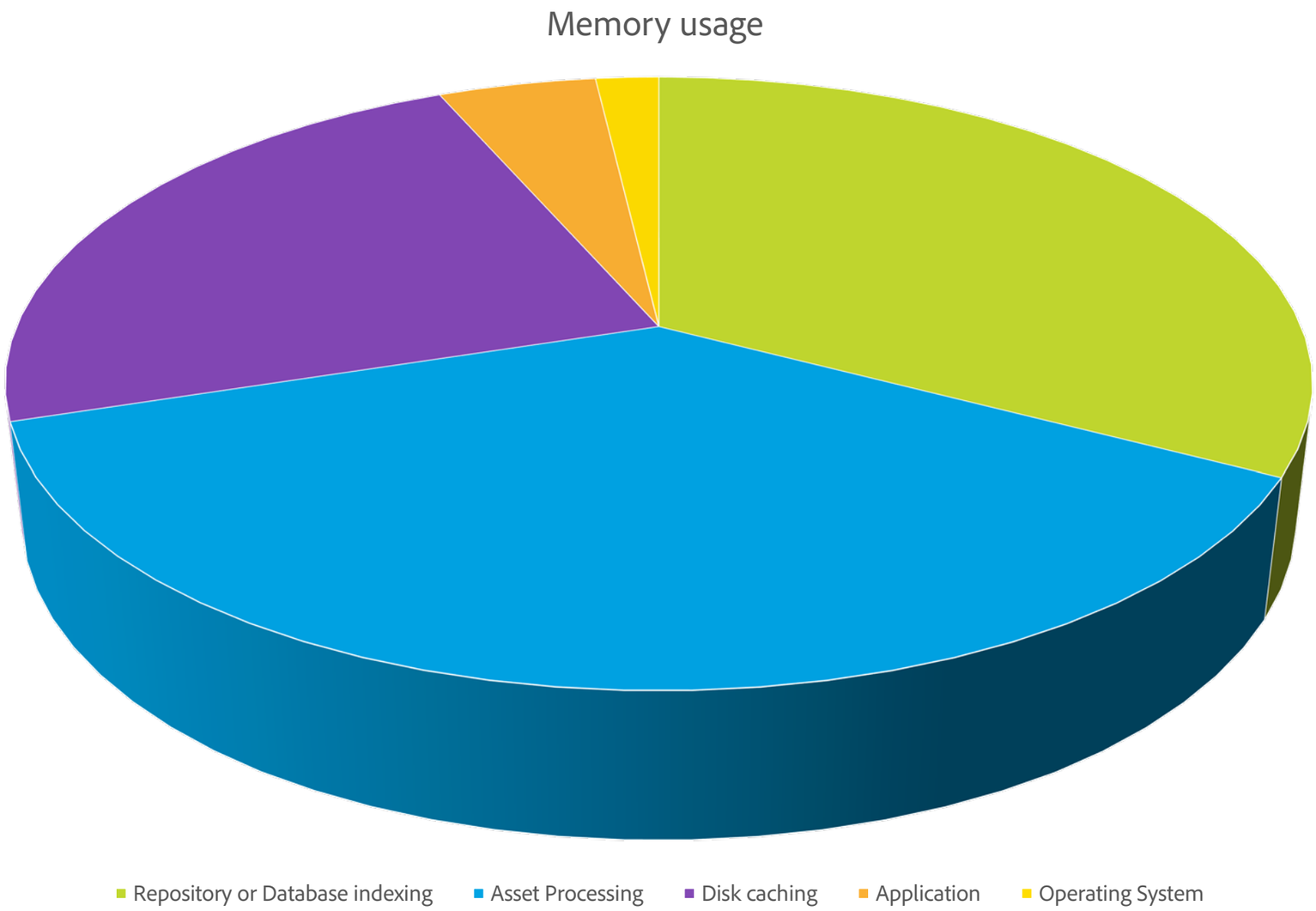


Adobe

What We Learned: More Server Disk I/O = More Performance



What We Learned: More Server RAM = More Disk I/O Performance



What We Learned: Bottlenecks and Priority

Priority

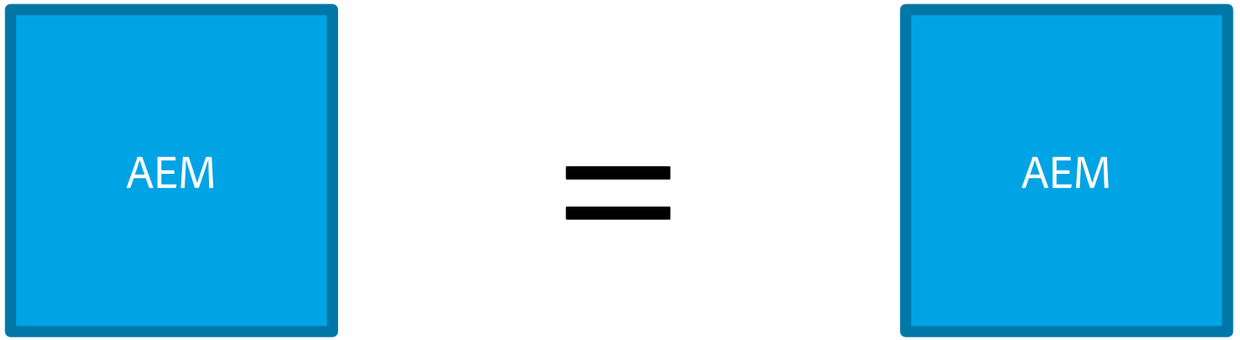
1. LAN bandwidth, including WiFi
2. Server Disk IO

   Individual spinning hard disks are not useful for large deployments.
3. Server RAM

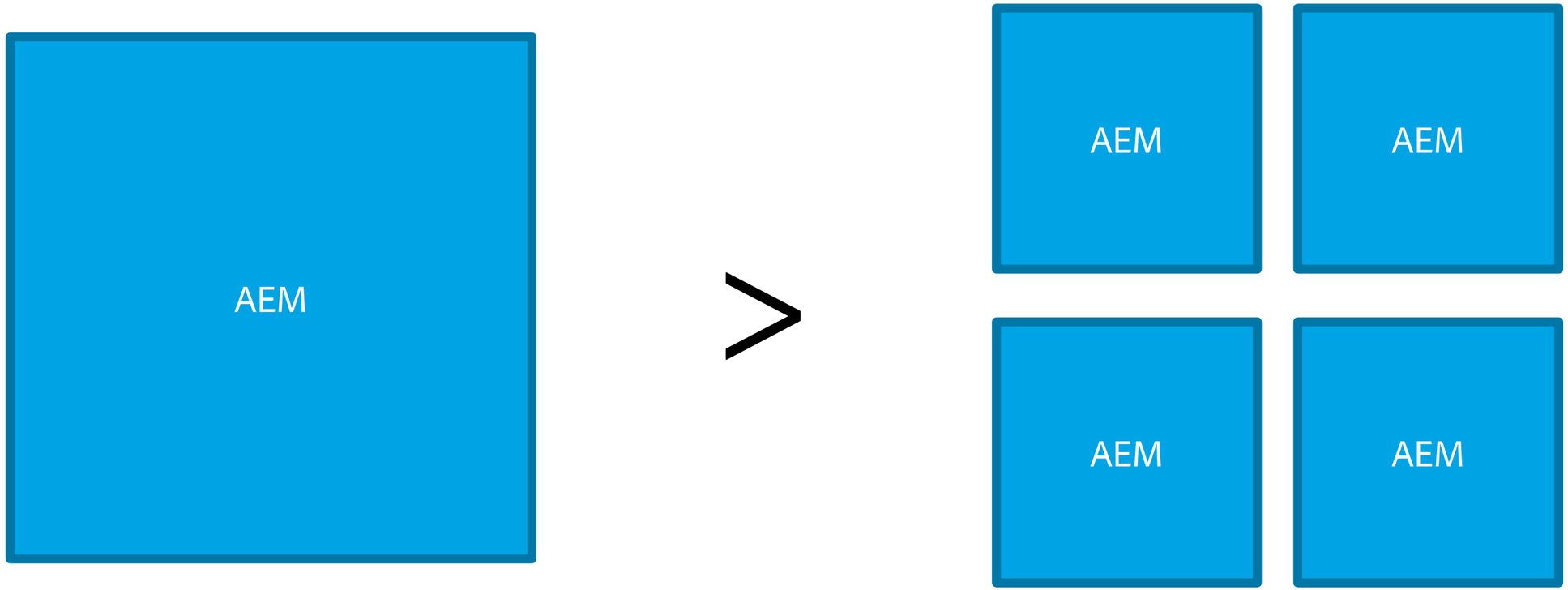
   Unused RAM improves Disk IO
4. Server CPU

Adobe

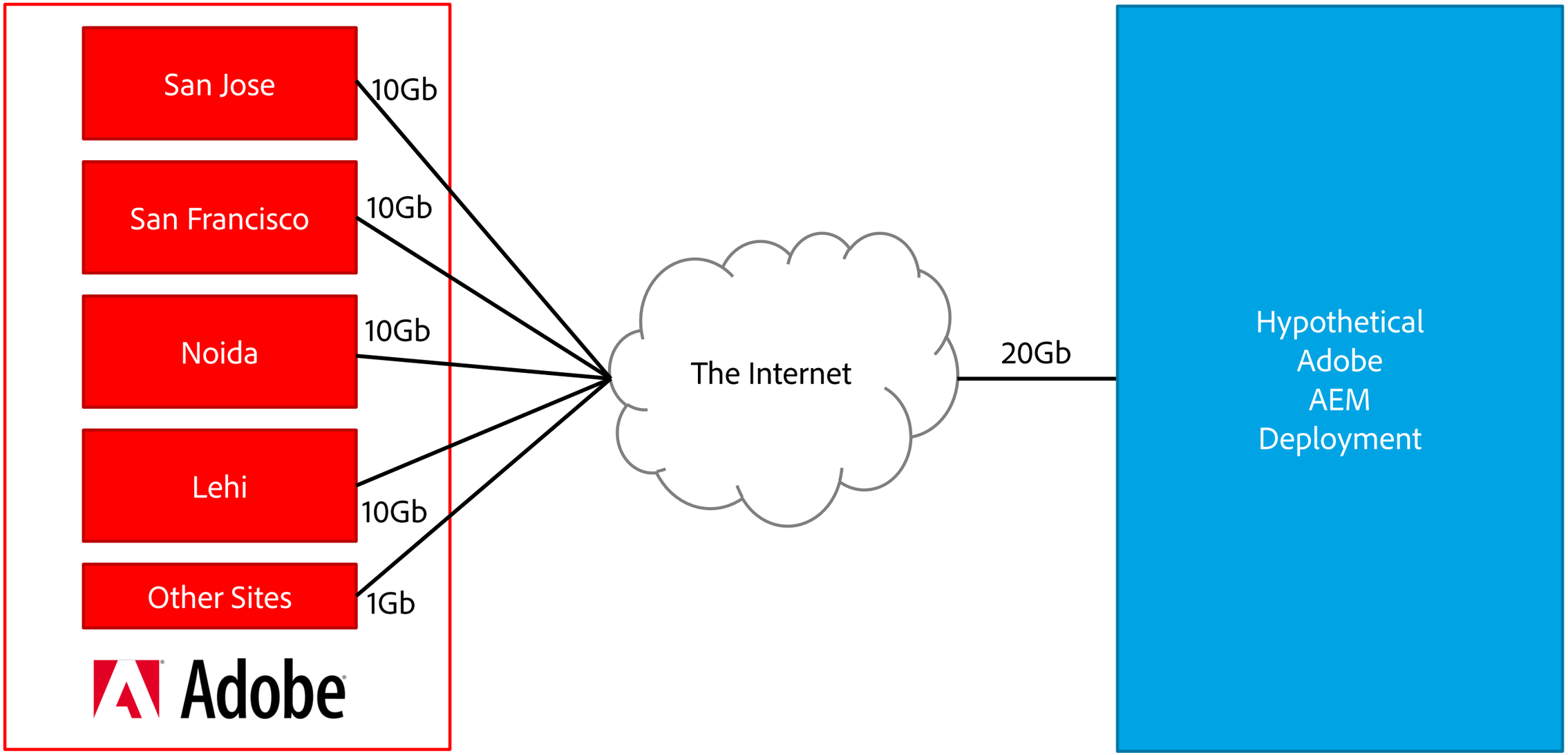
What We Learned: Vertical Scaling – More Effective than Horizontal Scaling



What We Learned: Vertical Scaling – More Effective than Horizontal Scaling



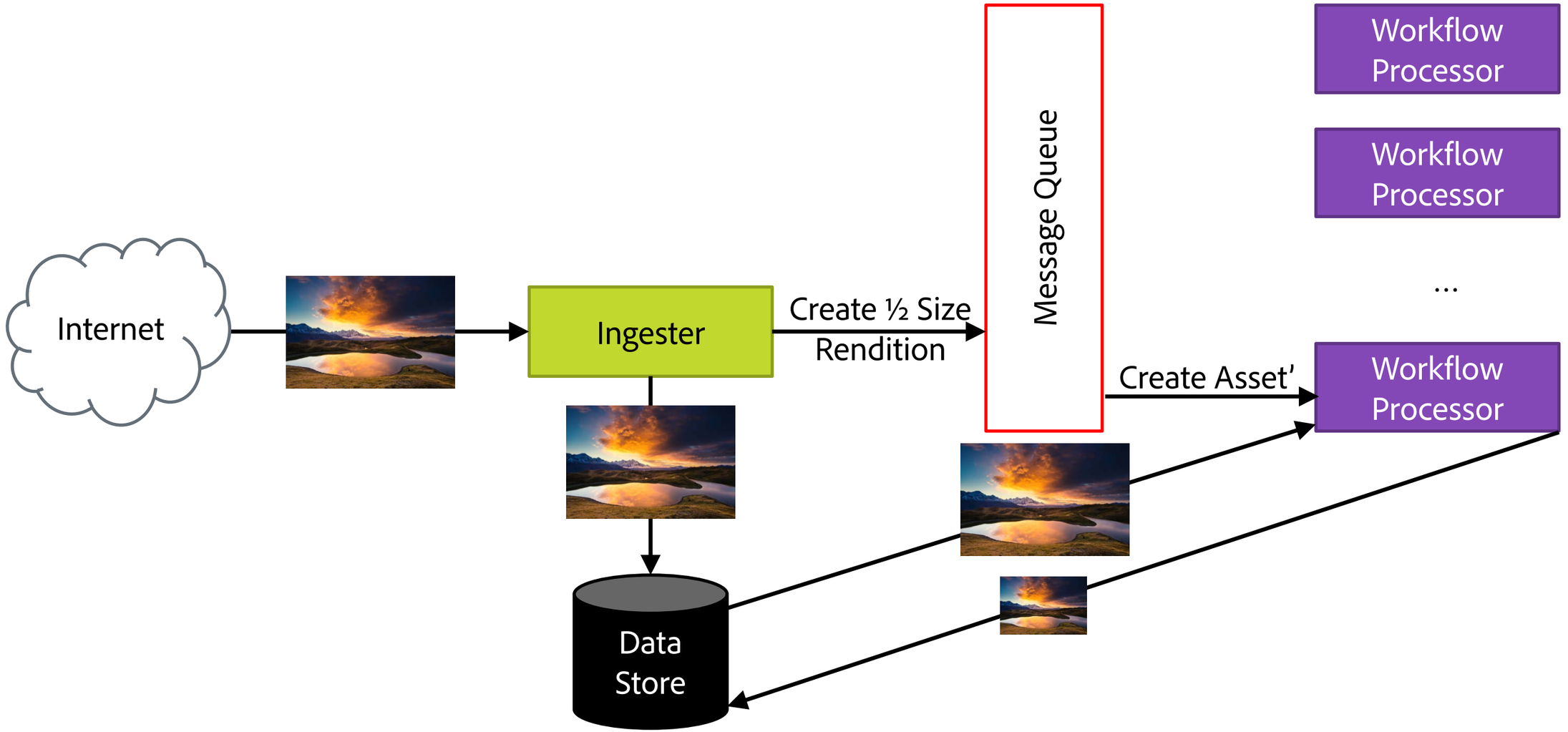
Can Vertical Scalability Really Be the Answer?



Takeaway

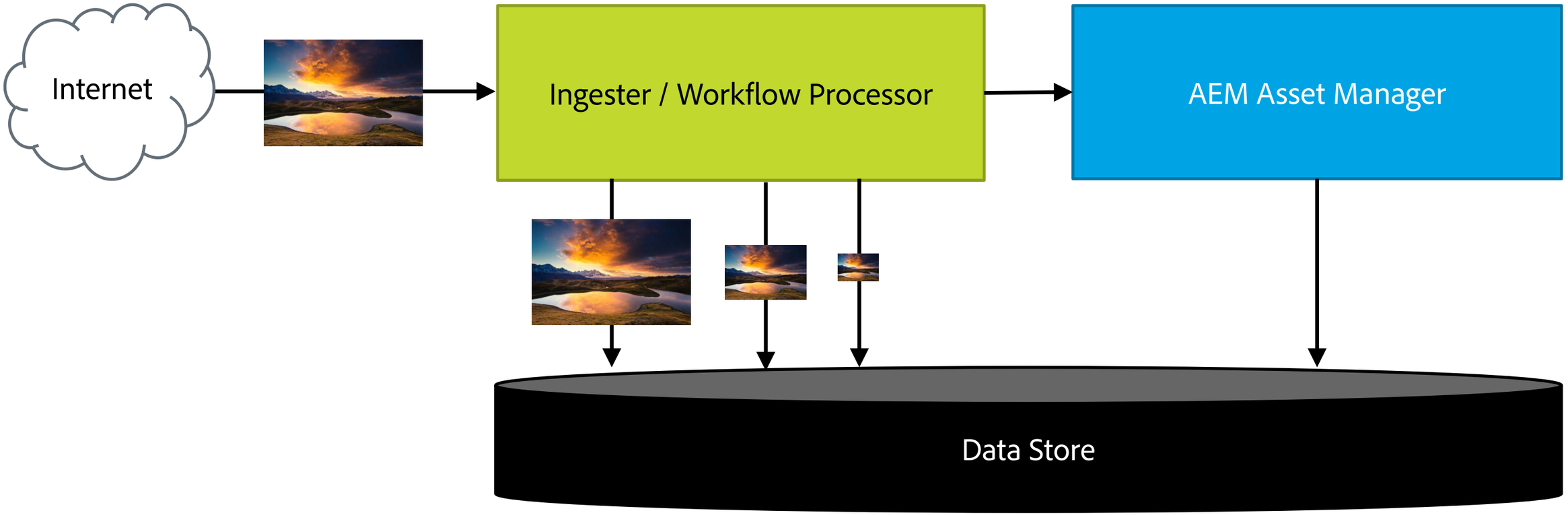
Sometimes, vertical scalability is the answer.

Separating Workflow Processing Reduces Performance



Adobe

Scaling the Ingestion Process



Takeaway

Don’t rely on assumptions.
Prove your assumptions before you invest in a major architectural redesign.

Adobe

Takeaway

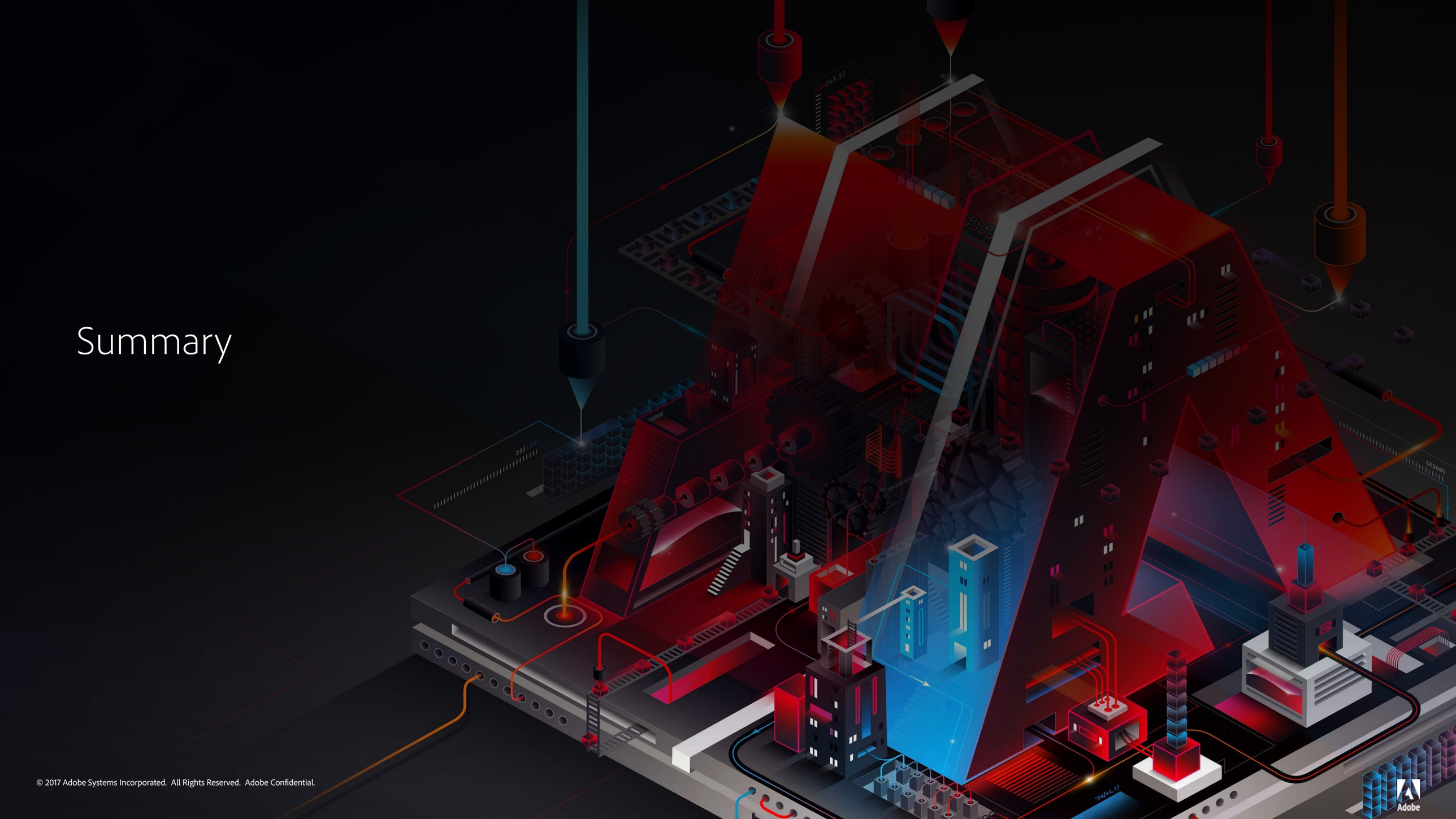
Take the time to understand the
best way to partition your application space.

Adobe

Scalability Challenge Number Three

The nature of the data may change the rules about scalability.

Summary

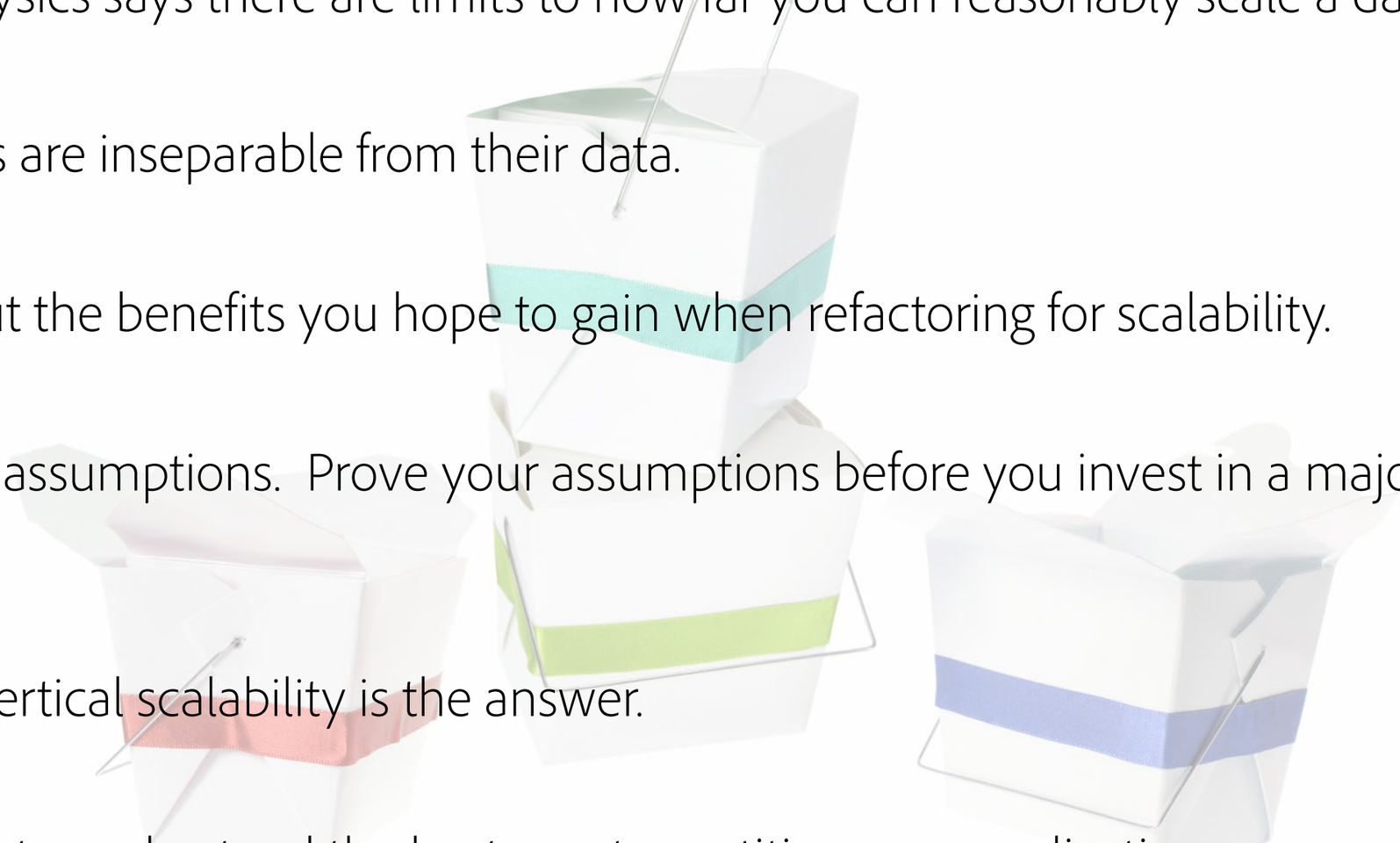


Three Scalability Challenges

- You cannot split a datastore into true microservices.
- LAN infrastructure may limit your ability to scale effectively.
- The nature of the data may change the rules about scalability.

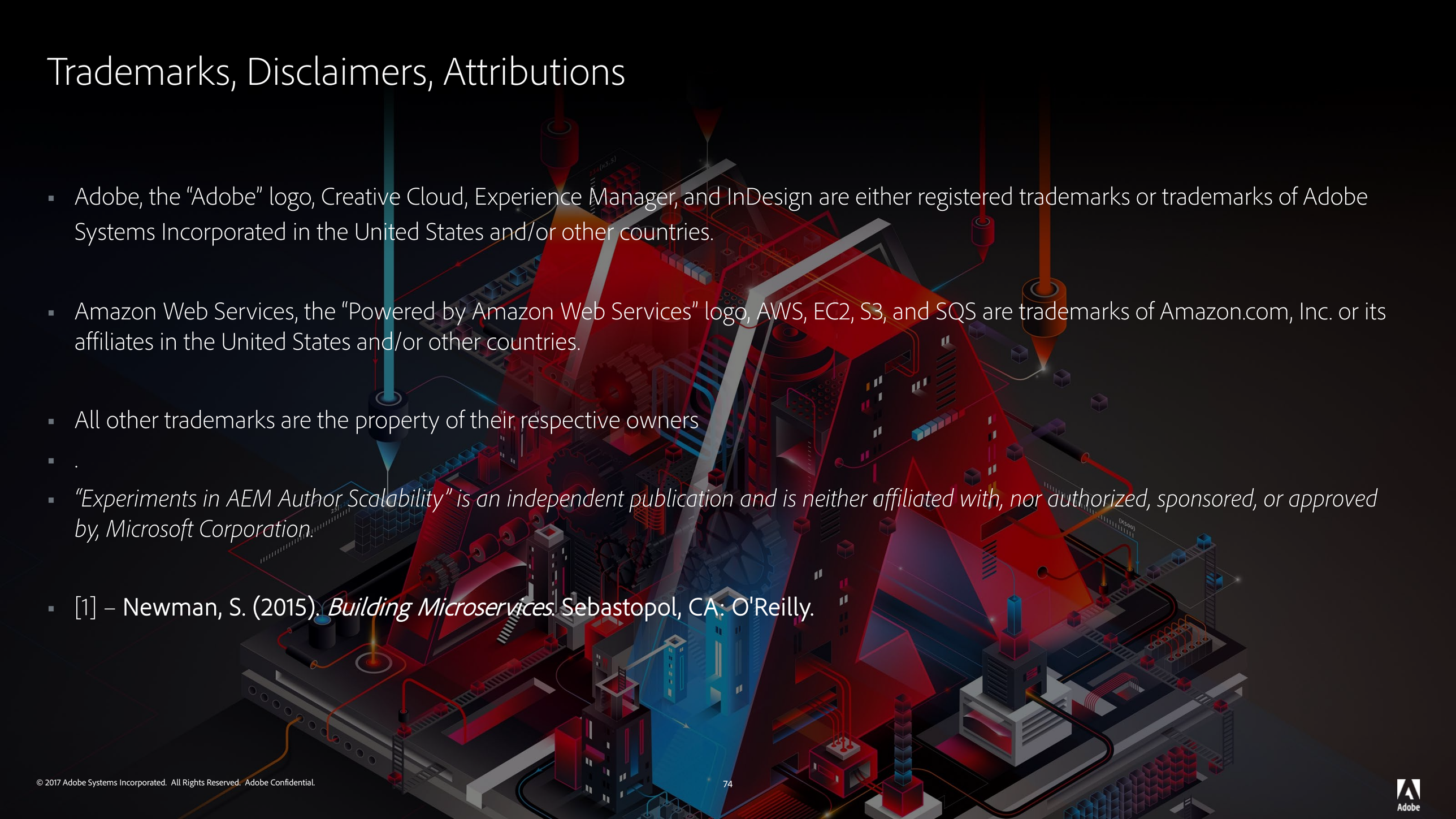
Takeaways

- Math and physics says there are limits to how far you can reasonably scale a database.
- Microservices are inseparable from their data.
- Be clear about the benefits you hope to gain when refactoring for scalability.
- Don't rely on assumptions. Prove your assumptions before you invest in a major architectural redesign.
- Sometimes vertical scalability is the answer.
- Take the time to understand the best way to partition your application space.



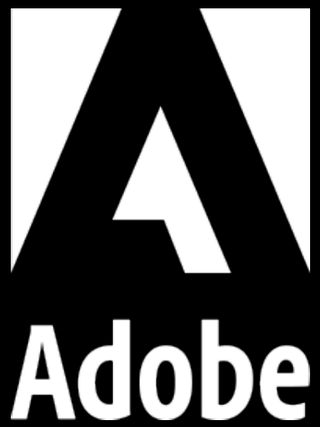
Adobe

Trademarks, Disclaimers, Attributions



- Adobe, the "Adobe" logo, Creative Cloud, Experience Manager, and InDesign are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Amazon Web Services, the "Powered by Amazon Web Services" logo, AWS, EC2, S3, and SQS are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.
- All other trademarks are the property of their respective owners
- .
- *"Experiments in AEM Author Scalability" is an independent publication and is neither affiliated with, nor authorized, sponsored, or approved by, Microsoft Corporation.*
- [1] – Newman, S. (2015). *Building Microservices*. Sebastopol, CA: O'Reilly.

Adobe


Adobe