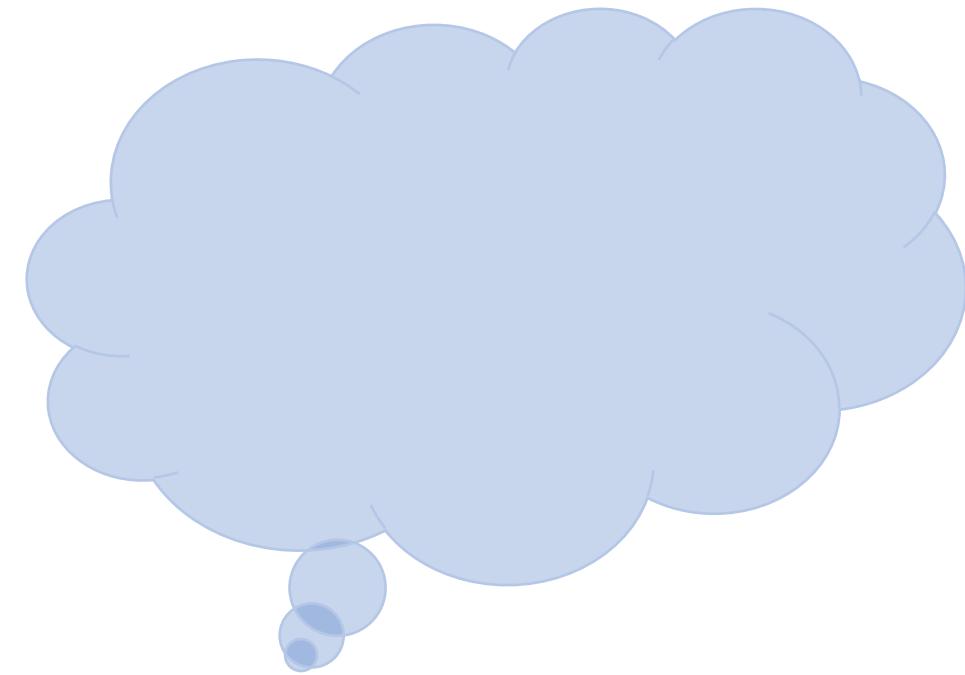


Building a Cloud Roadmap

Planning for a successful and efficient Cloud Migration

Course Topics

- Key challenges in moving to the cloud
- Examining your application
- Determine migration steps
- Objectives:
 - Understand common problems that occur in a migration
 - Understand risks and complications in making this happen
 - Understand & determine key performance metrics
 - Be able to create a plan/roadmap of your cloud migration journey



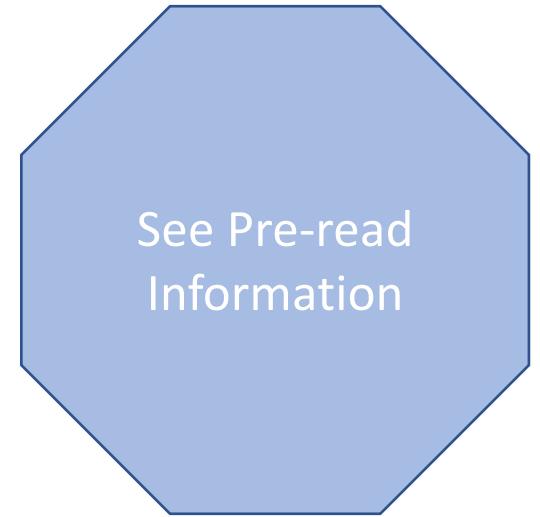
Target Audience

- IT Manager or Director in org responsible for cloud migrations
- Team has struggled or had only limited success
- Organization wants to modernize mission critical applications
 - > Cloud is considered an important step of this mission

Important Topics Not Covered

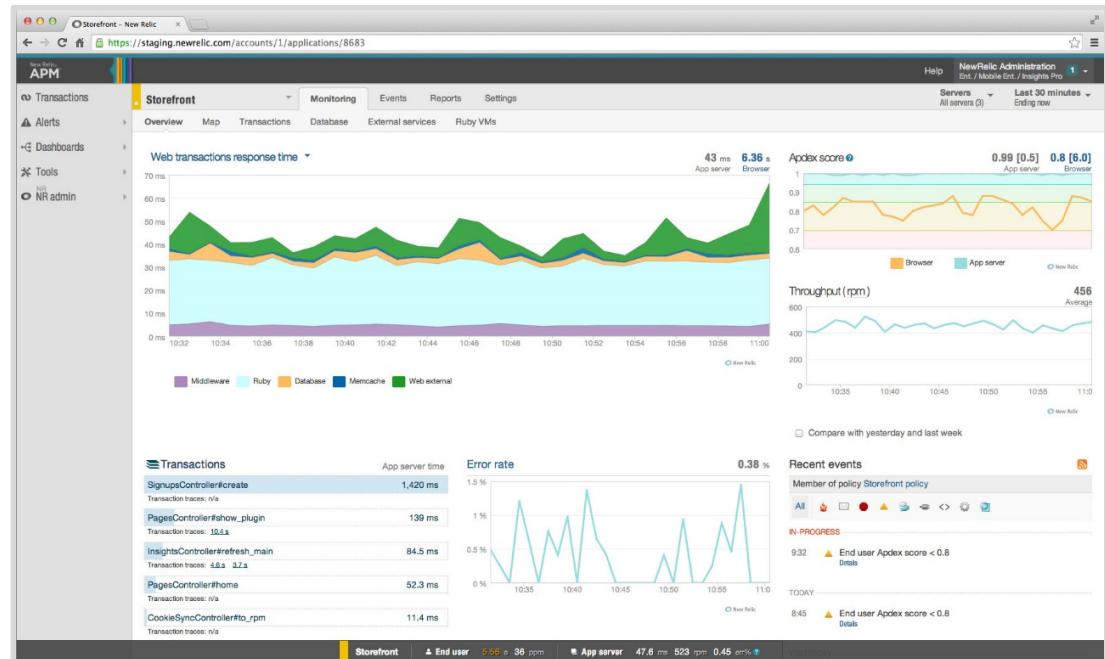
(covered in pre-read information)

- Services and microservices
- DevOps best practices
- CI/CD, Testing in Production, Quick Prototyping, Agile, ...
- Dynamic applications vs Static applications in the cloud
- Rearchitecting and modernizing applications
- Twelve factor applications
- Managing risk in projects



Performance Monitoring

- Understanding application/infrastructure performance
- Familiarity with APM (application performance monitoring) tools



Security and the Cloud



- Critical for all cloud migrations
- One of the most important steps in building a cloud environment
- Cloud providers have tooling that helps
- Cloud providers have expertise that helps
- Topic deserves significant attention
 - More attention than we can give it in this course
- See reference material for more information

Course Outline

- In** **Introduction** (10 minutes)
- Ev** **Initial Evaluation** (25 minutes)
- Pl** **Migration Planning** (20 minutes)
- Sc** **Scheduling Your Migration** (15 minutes)
- Pm** **Post Migration / Post Mortem** (10 minutes)
- QA** **Q&A** (10 minutes)

Introduction

- In** **Introduction** (10 minutes)
- Ev** **Initial Evaluation** (25 minutes)
- Pl** **Migration Planning** (20 minutes)
- Sc** **Scheduling Your Migration** (15 minutes)
- Pm** **Post Migration / Post Mortem** (10 minutes)
- QA** **Q&A** (10 minutes)

Two Migration Methods

Two Migration Methods

Shallow Migration

Deep Migration



Two Migration Methods

Shallow Migration

- App moved unchanged
- Cloud servers replicate on-prem
- Light integration
- “Lift-n-Shift”

Deep Migration

Static Cloud

Two Migration Methods

Shallow Migration

- App moved unchanged
- Cloud servers replicate on-prem
- Light integration
- “Lift-n-Shift”

Static Cloud

Deep Migration

- App updated to use cloud capabilities
- Utilize cloud-specific capabilities (Lambda)
- Cloud specific data stores (S3, DynamoDB)
- Deep integration

Dynamic Cloud

Two Migration Methods

Shallow Migration

- App moved unchanged
- Cloud servers replicate on-prem
- Light integration
- “Lift-n-Shift”

Static Cloud

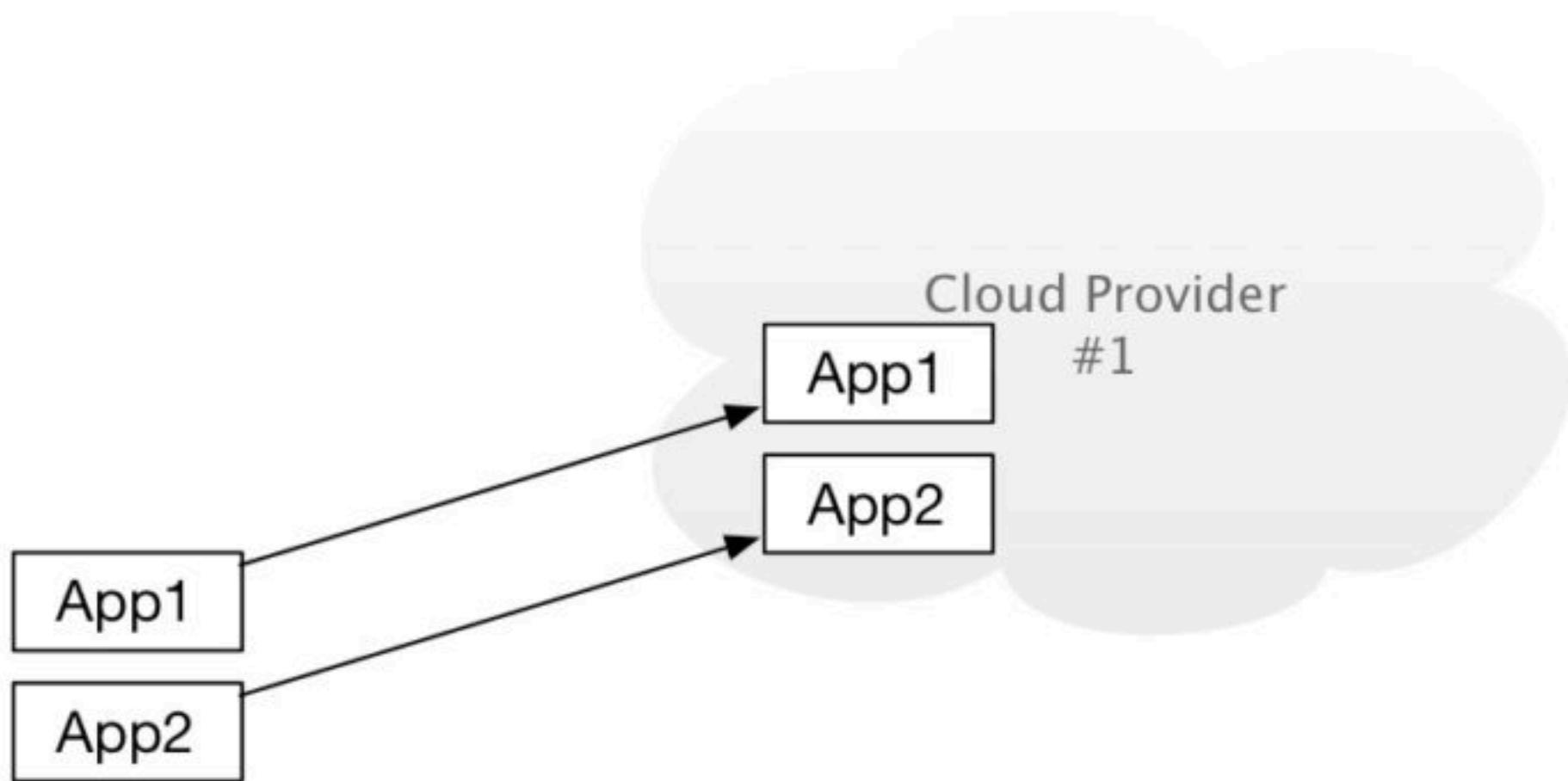
Deep Migration

- App updated to use cloud capabilities
- Utilize cloud-specific capabilities (Lambda)
- Cloud specific data stores (S3, DynamoDB)
- Deep integration

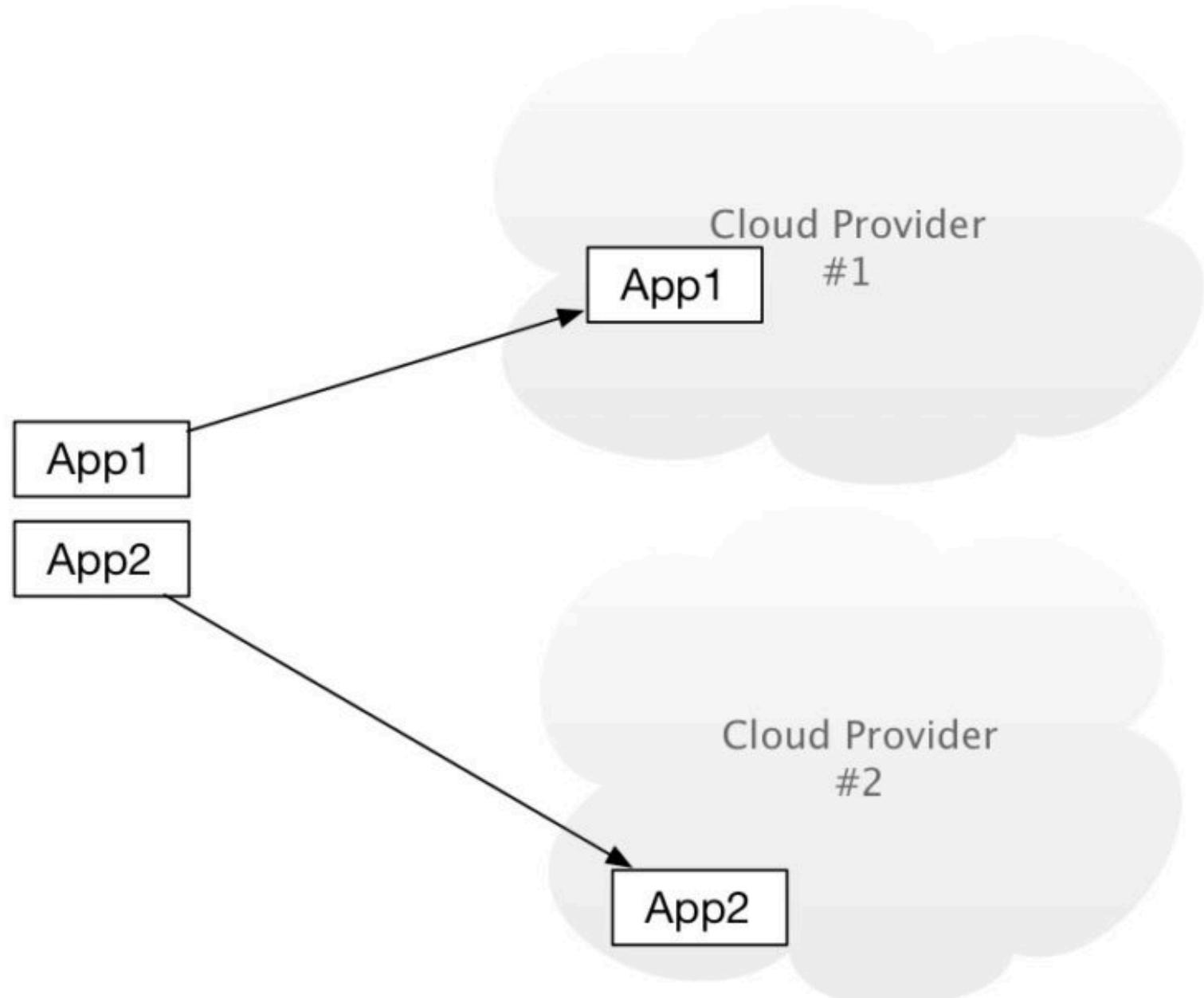
Dynamic Cloud

Single & Multi-Cloud

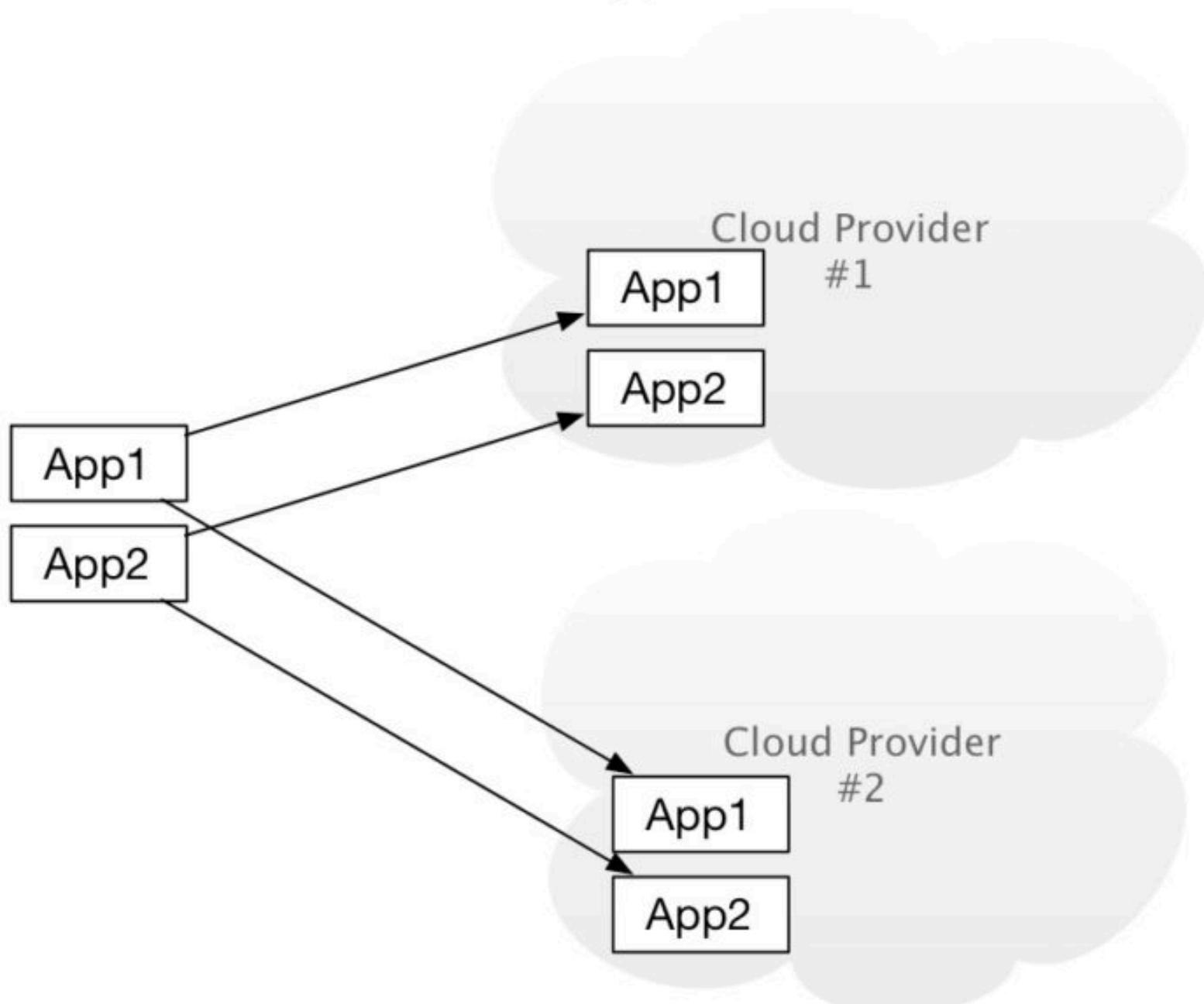
Single Cloud Application



Selective Cloud Application



Joint Cloud Application



What Model Best Describes Your Needs?

- Migration Method:
 - Shallow Migration / Static Cloud
 - Deep Migration / Dynamic Cloud
- Number of Clouds:
 - Single Cloud?
 - Multi-Cloud: Each Application/Service on one cloud?
 - Multi-Cloud: Each Application/Service on multiple clouds?

Initial Evaluation

- In Introduction (10 minutes)
- Ev **Initial Evaluation** (25 minutes)
- Pl Migration Planning (20 minutes)
- Sc Scheduling Your Migration (15 minutes)
- Pm Post Migration / Post Mortem (10 minutes)
- QA Q&A (10 minutes)

Initial Evaluation - Level of Analysis

- We're Going to do an initial evaluation for your application
- Two parts:
 1. High level evaluation for entire application
 2. Component level evaluation for each component
- Process is basically the same
 - Some differences, we'll discuss during the analysis

KPIs

KPIs

- Key Performance Indicators
- Examples:
Error Rate, Response Time, Throughput, Availability
- Key to evaluating effectiveness of your application
- We use for:
 - Baselining and Success Criteria
 - Evaluation and Diagnosis

KPIs Help Show Success

- During Migration:
 - Are there lurking problems within your application?
- After Migration:
 - Was the migration completed?
 - Was your migration successful?

Characteristics of Useful KPIs

- Good KPIs Can Be:
 - Indicative of the customer experience
 - Early indicator of an internal problem
 - Important indicator of business success

Establishing KPIs

- General process:
 - Brainstorm metrics
 - Categorize by Type
 - Filter and prioritize
 - Determine acquisition method
 - Determine facets

KPIs: Brainstorm List

- Brainstorm a list of possible KPIs:
 - Base list provided
 - Industry experience
 - Metrics company uses regularly
 - Personal experience
- Think about:
 - External vs Internal
 - Customer centric & system centric
 - Mobile/Browser & Application & Datastore & external services

Example KPIs

- Apdex
- Availability
- Throughput
- Response Time
- Error %
- Session counts
- Session duration
- Webpage Name
- Page views
- Page load time
- Memory Footprint
- CPU Workload
- Disk Workload
- Network Throughput
- Cart Adds
- Cart Conversions
- Ad engagement rate

KPIs: Categorize

Categories for KPIs:

- User Experience
 - Page load time, Lag, Response time, ...
- Application/Component Performance
 - Error rates, Throughput, ...
- Infrastructure
 - CPU, Disk Perf, Memory Usage, Network Throughput, ...
- Business
 - Cart Adds, Conversions, Engagement Rates, ...

KPIs: Filter and Prioritize

Determine most important KPIs:

1. Most important to business
 - Typical categories: *business* or *user experience*
 2. Potential for most impacted by migration
 - Typical categories: *component performance* or *infrastructure*
- Prioritize all in order:
 - *Top ones/most critical/most impactful* ← KPIs
 - *Useful for diagnosing problems* ← Still important
 - *Nice to know*
 - *Uncertain usefulness*

KPIs: Acquisition Method

How are you going to measure the KPI?

- Synthetic Load Test
- Browser instrumentation
- Application instrumentation (e.g. APM)
- Server instrumentation

Facets

- Way to categories for filtering
- Examples:
 - User characteristics
 - Geography
 - Device Type
 - OS Type
 - Carrier
 - Browser Type
 - System characteristics
 - Server(s) utilized

Worksheet – Determining KPIs

KPIs: Brainstorm List

- Brainstorm a list of possible KPIs:
 - Base list provided
 - Industry experience
 - Metrics company uses regularly
 - Personal experience
- Think about:
 - External vs Internal
 - Customer centric & system centric
 - Mobile/Browser & Application & Datastore & external services

Example Metrics:

- Apdex
- Availability
- Throughput
- Response Time
- Error %
- Session counts
- Session duration
- Webpage Name
- Page views
- Page load time
- Memory Footprint
- CPU Workload
- Disk Workload
- Network Throughput
- Cart Adds
- Cart Conversions
- Ad engagement rate

Worksheet: Determine KPIs	
Application/Service: _____	
Brainstorm Metrics	
<p>KPIs: Brainstorm Process Brainstorm a list of possible KPIs:<ul style="list-style-type: none">• Base list provided• Industry experience• Metrics company uses regularly• Personal experienceThink about:<ul style="list-style-type: none">• External vs Internal• Customer centric & system centric• Mobile/Browser & Application & Datastore & external services</p>	
Categorize Metrics	
User Experience	Component Performance
Infrastructure	Business

Worksheet: Categorize

- User Experience
- Component Performance
- Infrastructure
- Business

Worksheet: Determine KPIs

Application/Service: _____

Brainstorm Metrics

KPIs: Brainstorm Process
Brainstorm a list of possible KPIs:

- Base list provided
- Industry experience
- Metrics company uses regularly
- Personal experience

Think about:

- External vs Internal
- Customer centric & system centric
- Mobile/Browser & Application & Datastore & external services

Categorize Metrics

User Experience	Component Performance
Infrastructure	Business

Worksheet: Filter and Prioritize

Identify most important KPIs

1. Business
2. Impacted by migration

Worksheet: Determine KPIs

Application/Service: _____

Brainstorm Metrics

KPIs: Brainstorm Process
Brainstorm a list of possible KPIs:

- Base list provided
- Industry experience
- Metrics company uses regularly
- Personal experience

Think about:

- External vs Internal
- Customer centric & system centric
- Mobile/Browser & Application & Datastore & external services

Categorize Metrics

User Experience	Component Performance
Infrastructure	Business

Worksheet: Acquisition Method

How are you acquiring?

1. Synthetic
 2. Browser Instrumentation
 3. Application Instrumentation
 4. Server Instrumentation

Reference Information

<u>Example KPIs</u>	<u>Example KPI Facets</u>	<u>Categories</u>
Apdex	Geography	<ul style="list-style-type: none"> User experience
Availability	Device Type	<ul style="list-style-type: none"> Component Performance
Throughput	OS Type	<ul style="list-style-type: none"> Infrastructure
Response Time	Carrier	<ul style="list-style-type: none"> Business
Error %	Browser Type	
Session counts		
Session duration		
Webpage Name		
Page views		
Page load time		
Memory Footprint		
CPU Workload		
Disk Workload		
Network Throughput		
Cart Adds		
Cart Conversions		
Ad engagement rate		

Baselining

Baselining

- Analyzing current (pre-migration) performance in order to determine acceptable future (post-migration) performance
- Used to:
 - Determine when migration is complete (am I done yet?)
 - Diagnose migration problem (what's changed?)
 - Validate post-migration improvements (what have I accomplished?)

Baseline Strategy

- How long do I examine current performance to determine baseline?
 - One Day?
 - One Week?
 - One Month?
- Average? Min/Max? What period? {15 minute running average}
- What about variant performance?
 - Is period selected representative of actual performance?
 - How variable is the data?
- What's More Important?
 - Steady state?
 - Exceptional spikes?
- Need to determine baseline strategy ahead of time

Baseline Determination

- Determine strategy (day/week/month, avg/min/max, normal/exceptional)
- Collect data

More Information: Chapter 18, "Service Level Agreements", Architecting for Scale, Lee Atchison
Available in O'Reilly Safari Books

Baseline Determination

- Once you've collected the data:
 - Determine expected range, desired range, acceptable range
 - {from that}Determine acceptance criteria

Example: Service Latency Baseline

- Determine strategy:
 - gather data over seven day period
 - 15 minute average, TP90
- Collected data:
{examples: lowest day: 250ms, TP90 350ms, highest day: 270ms, TP90 550ms}
- Determine:
 - Expected Range: 250ms-270ms, TP90: 350ms-550ms
 - Desired Range: 240ms-260ms, TP90: 300ms-450ms
 - Acceptable Range: <270ms, TP90 <550ms
- Acceptance Criteria:
3 consecutive days where average in acceptable range 90% of the time
- *Question: Do you feel you could create baselines for your migrations?*

Migration Planning

- In Introduction (10 minutes)
- Ev Initial Evaluation (25 minutes)
- Pl **Migration Planning** (20 minutes)
- Sc Scheduling Your Migration (15 minutes)
- Pm Post Migration / Post Mortem (10 minutes)
- QA Q&A (10 minutes)

Migration Architect

- System architect level position (senior architect)
- Responsible for all aspects of completing the migration:
 - Defining refactorings
 - Data migration strategies
 - Cloud solution requirements
 - Migration priorities
 - Production switchover planning
- Specific duties discussed during this class

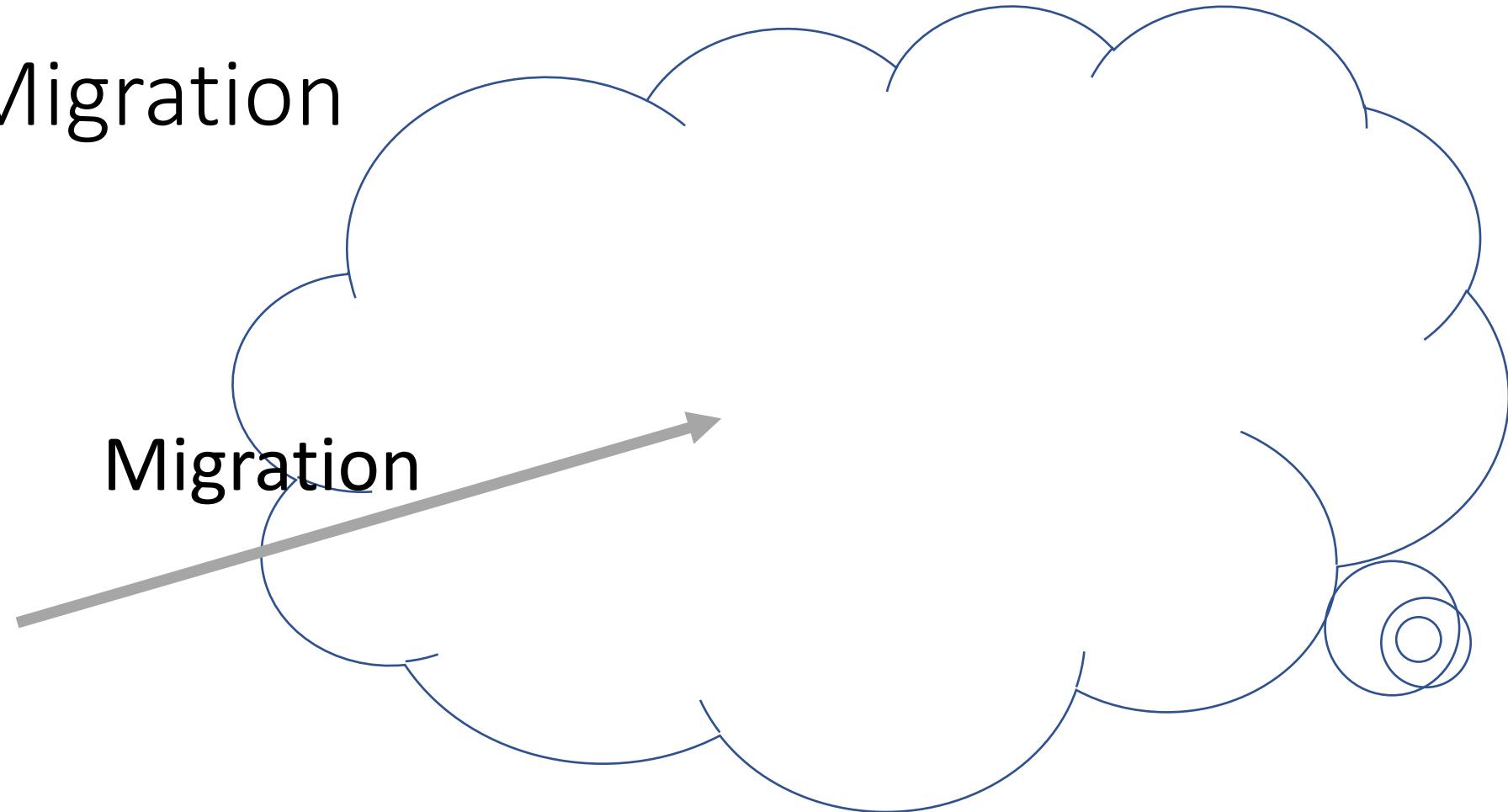
Prioritize Migration Components

Monolith Migration

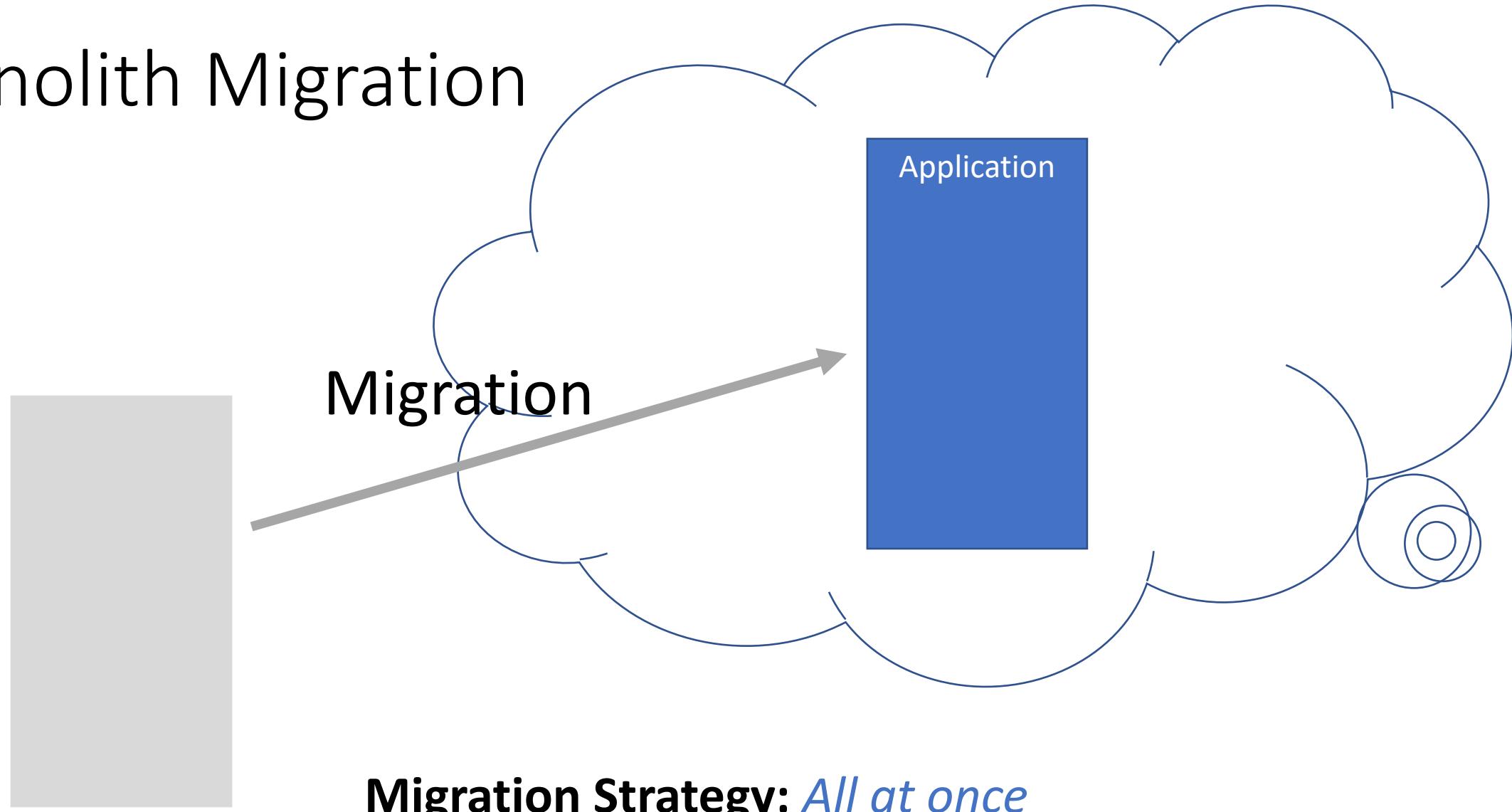


Migration

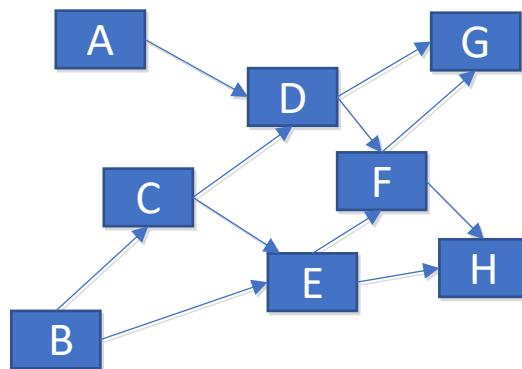
Migration Strategy: *All at once*



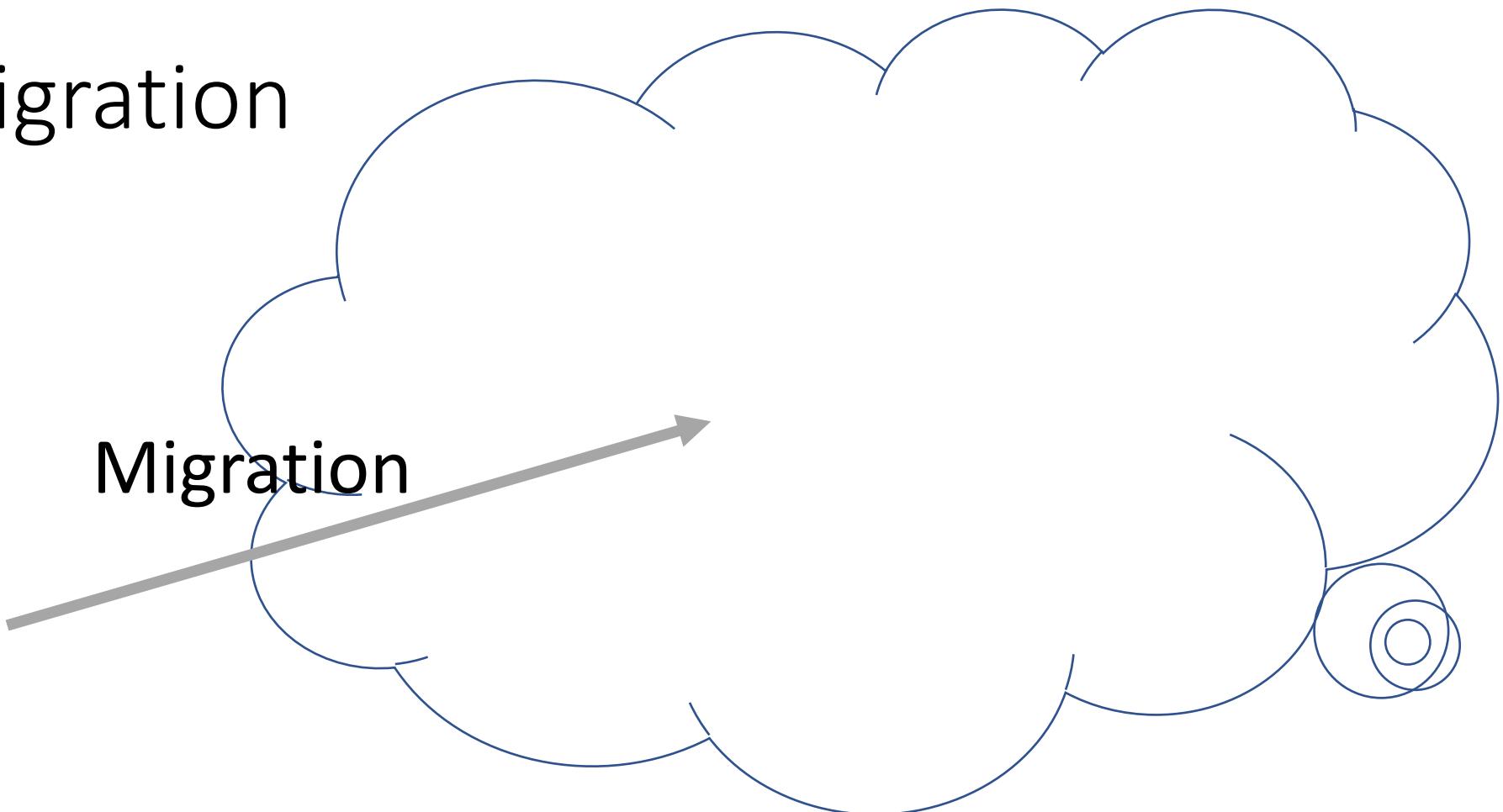
Monolith Migration



Services Migration

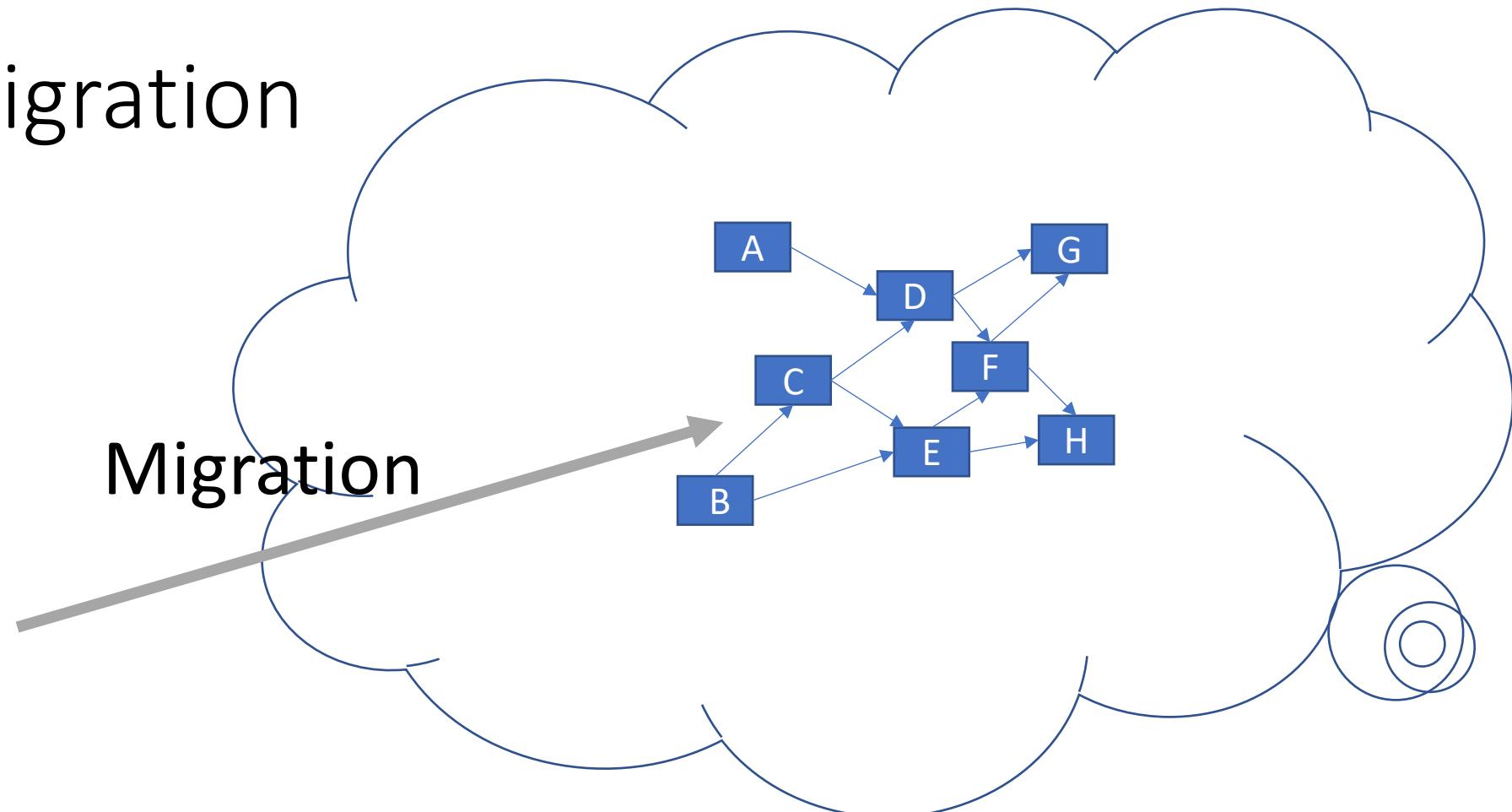
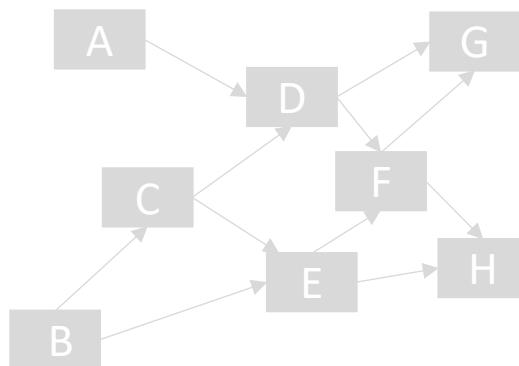


Migration



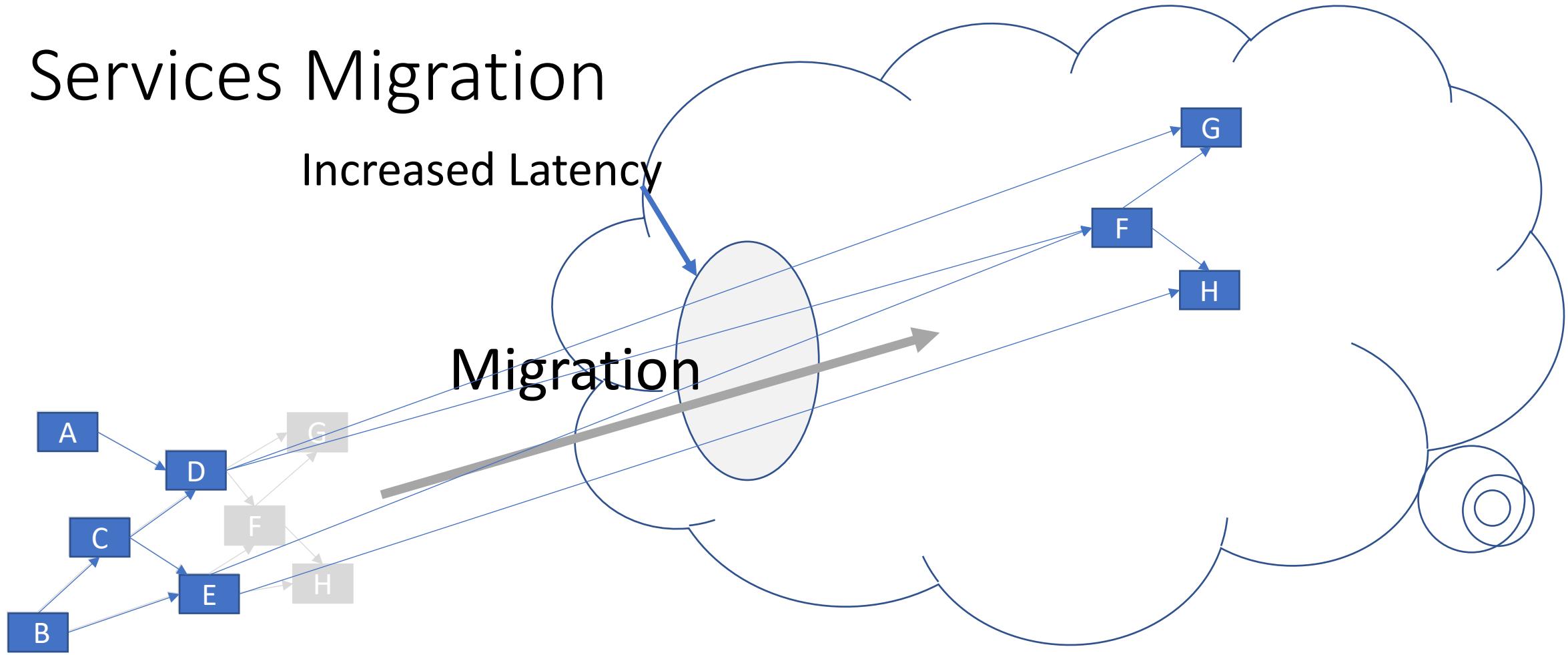
Migration Strategy: *All at once*

Services Migration



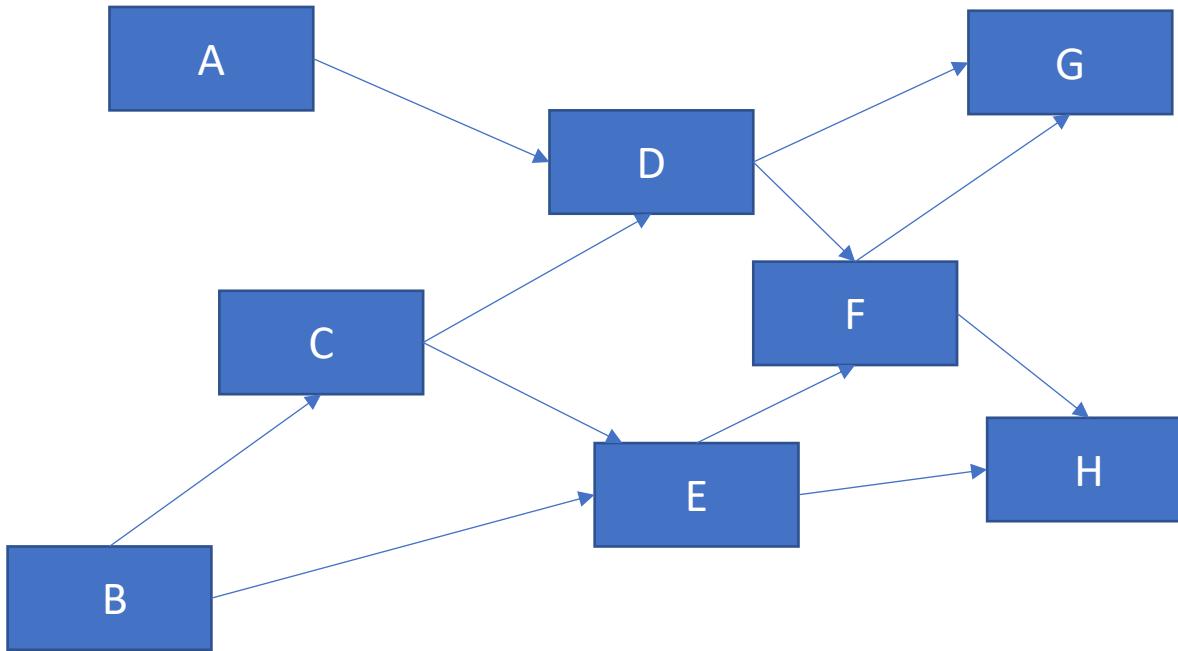
Migration Strategy: *All at once*

Services Migration



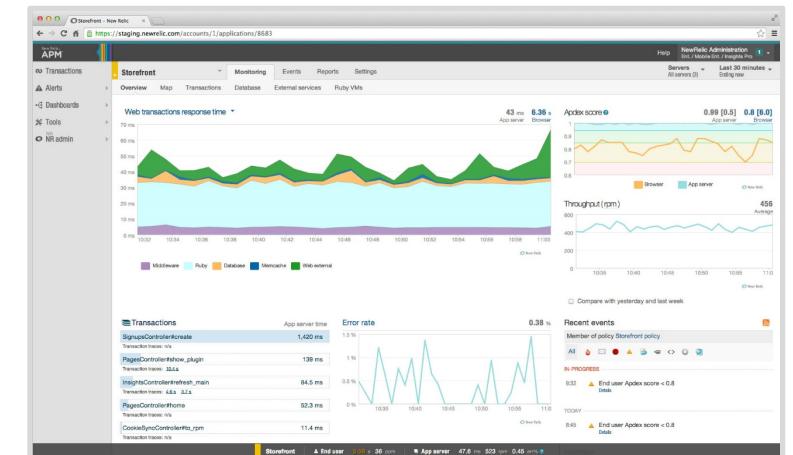
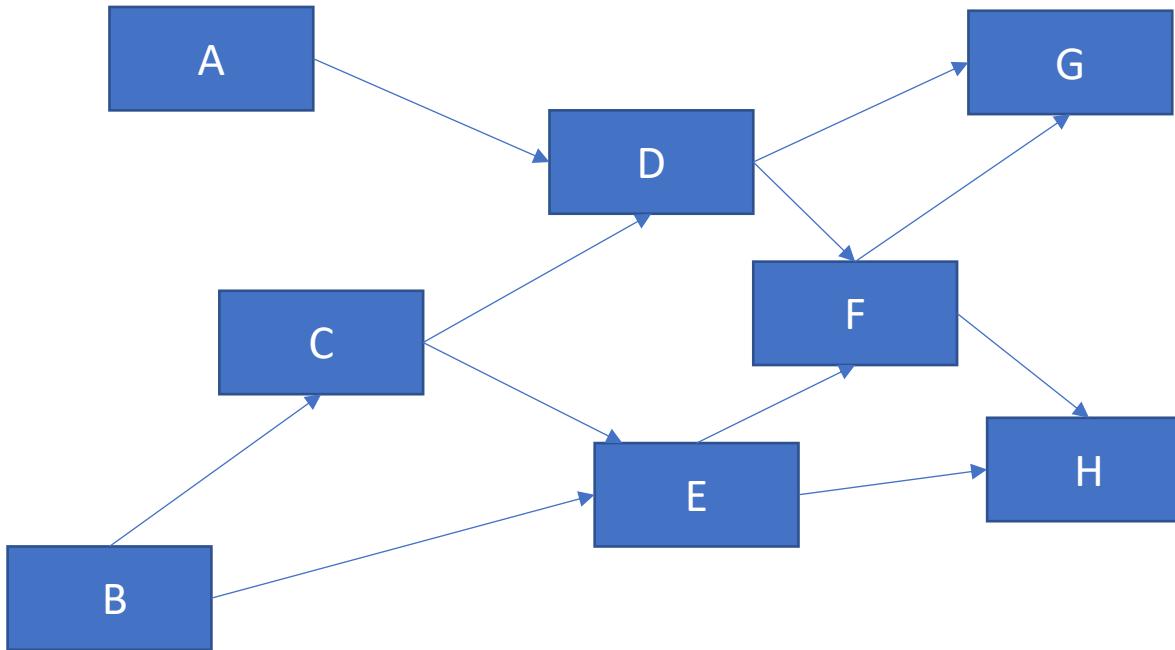
Migration Strategy: *One Service at a Time*

Service Map



- Services
- Standalone data stores
- External applications
- External APIs

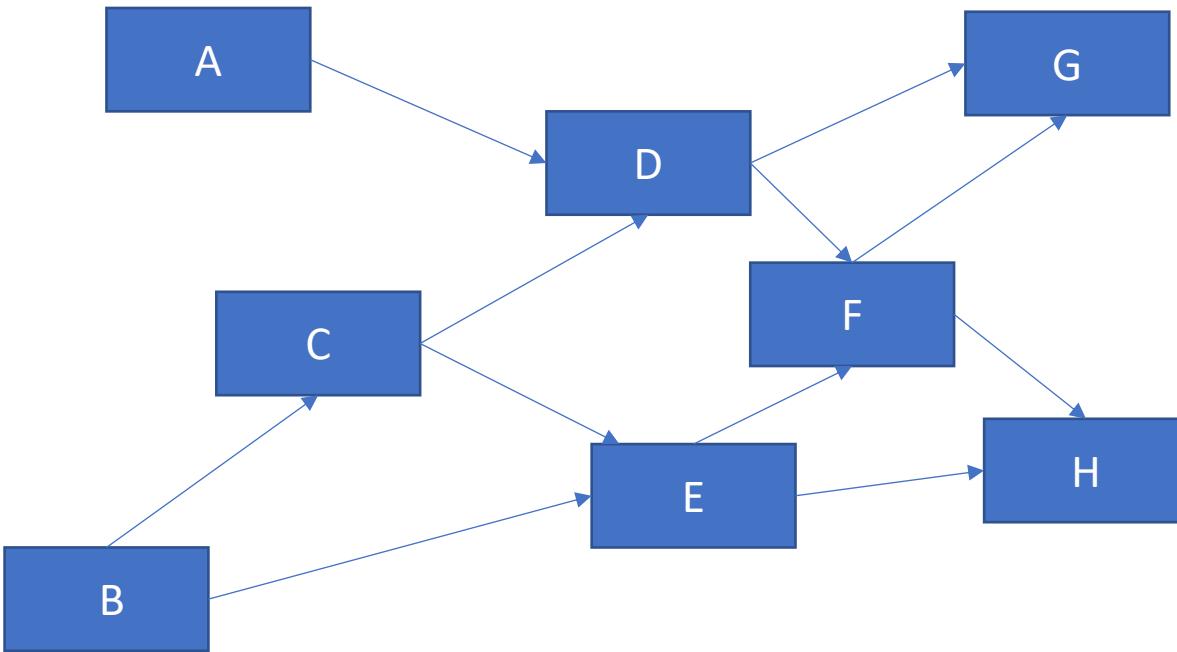
Service Map



Application Performance Management

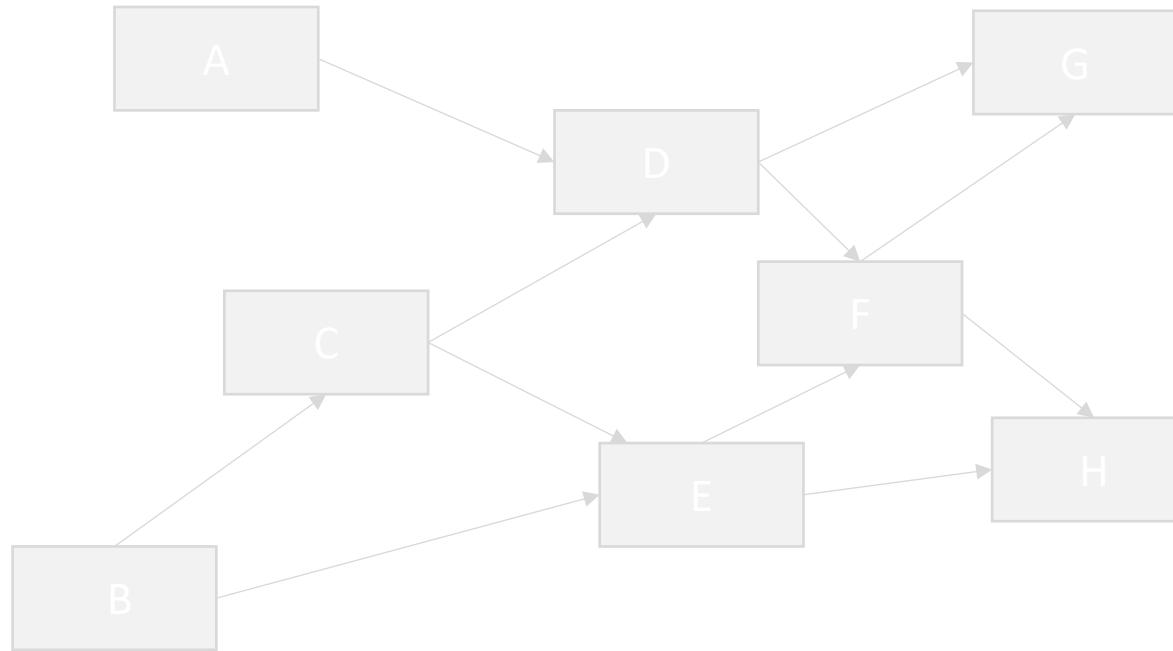
Service Map

- Inside Out
- Outside In



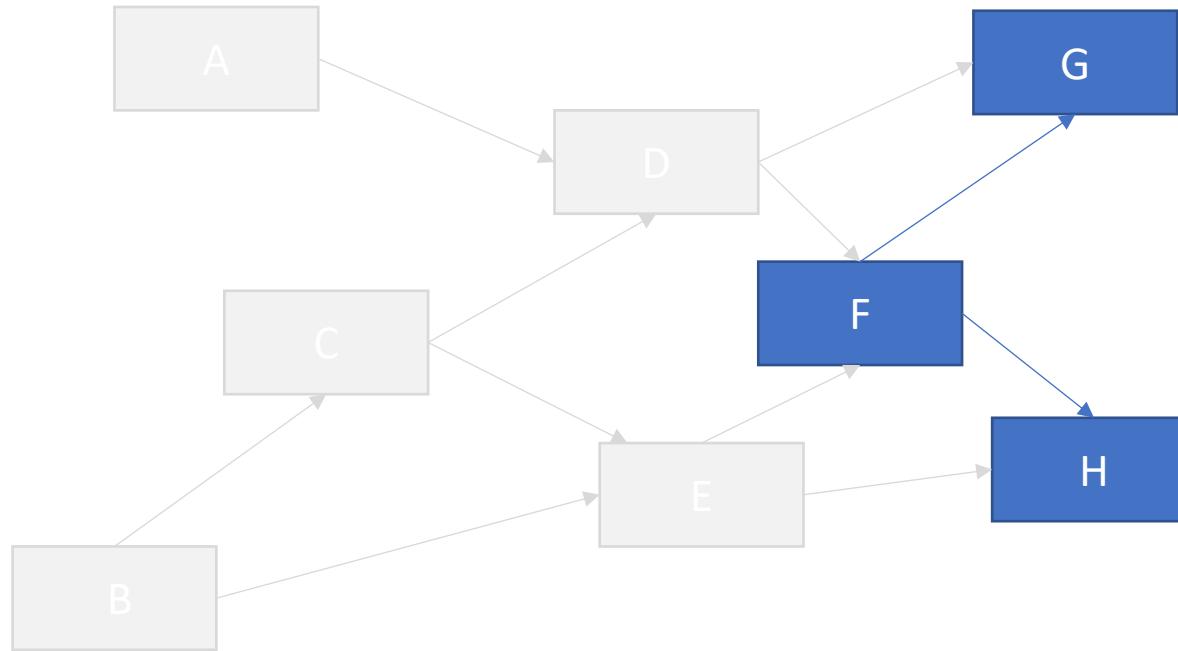
Inside Out Migration

Users



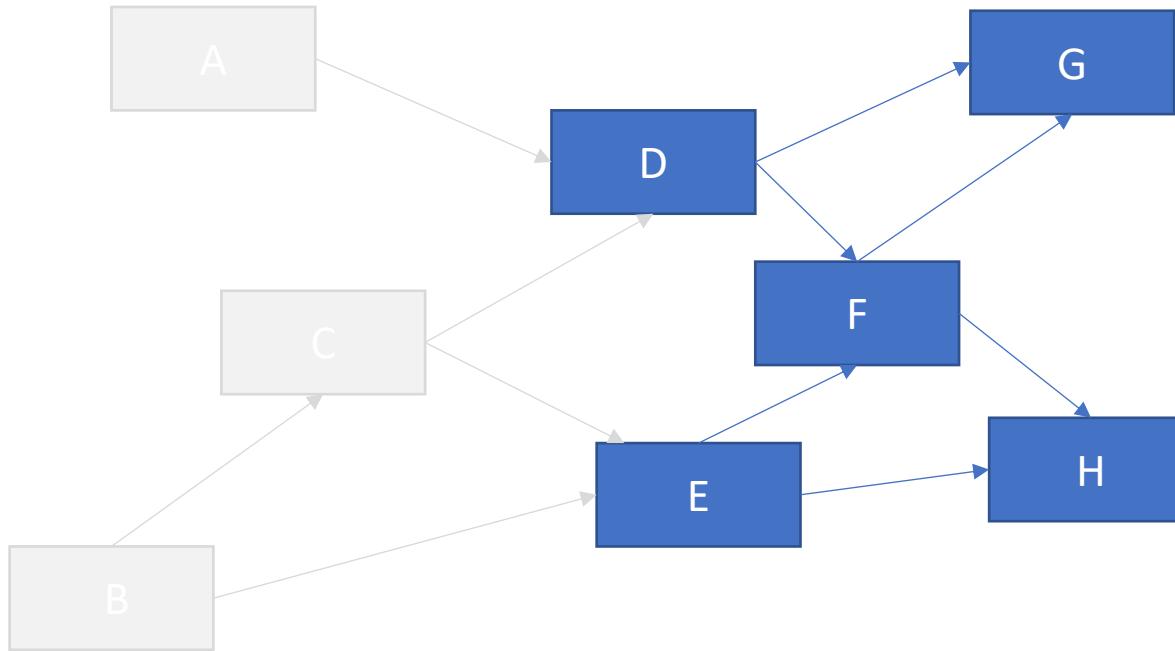
Inside Out Migration

Users



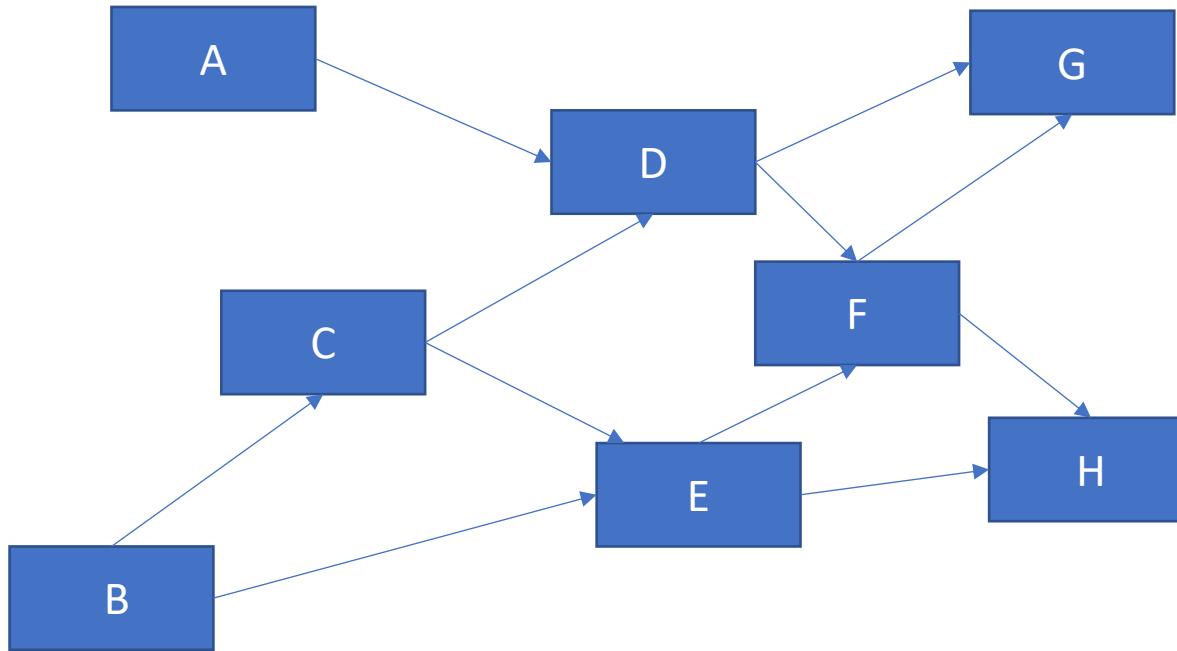
Inside Out Migration

Users

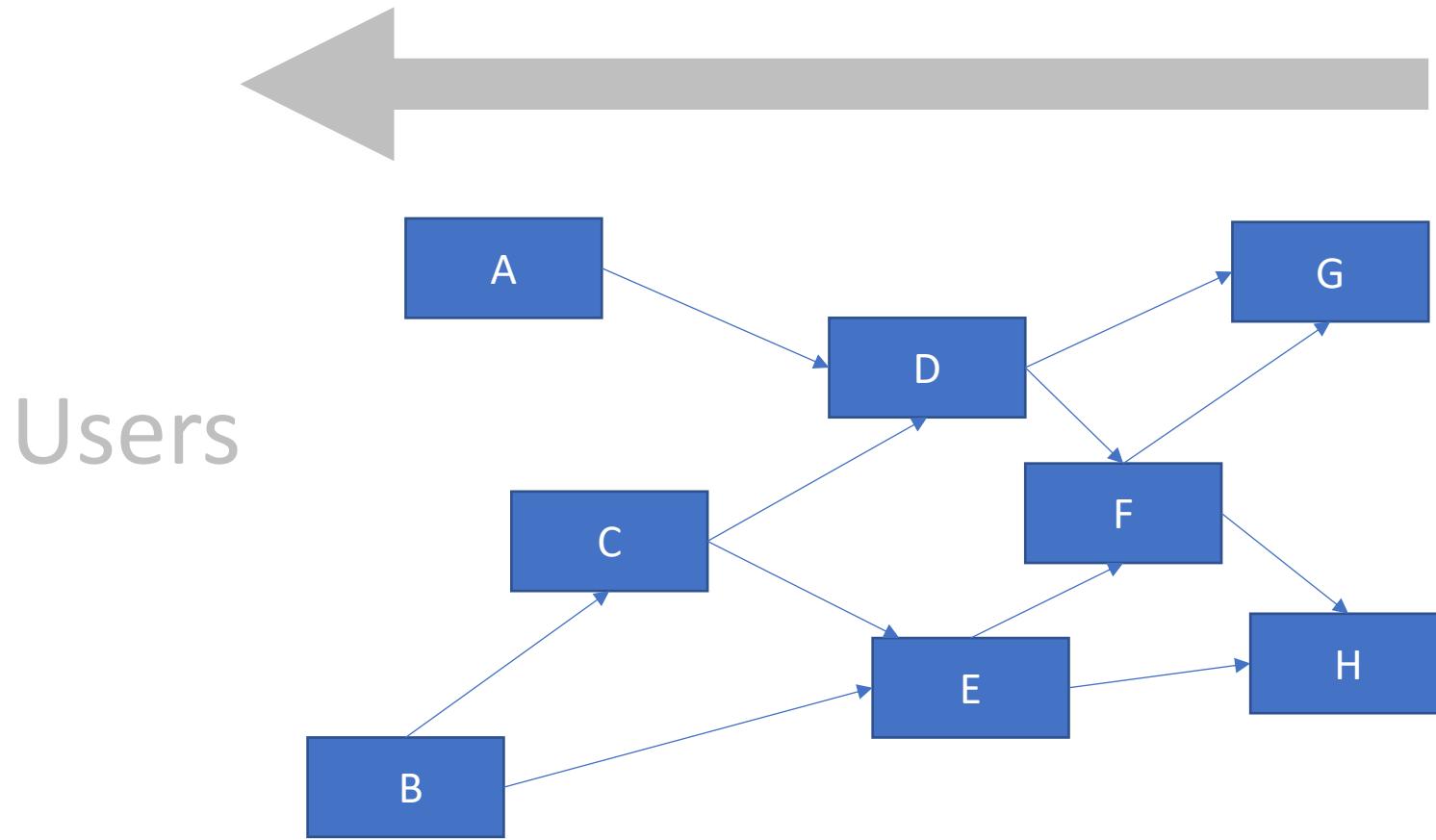


Inside Out Migration

Users

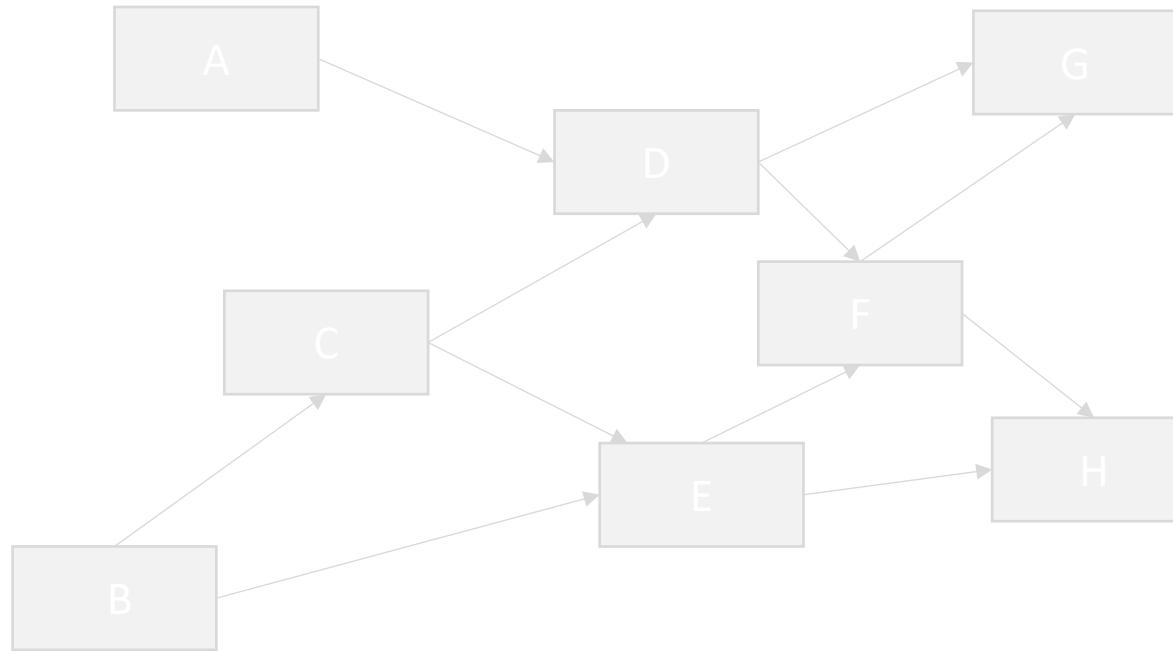


Inside Out Migration



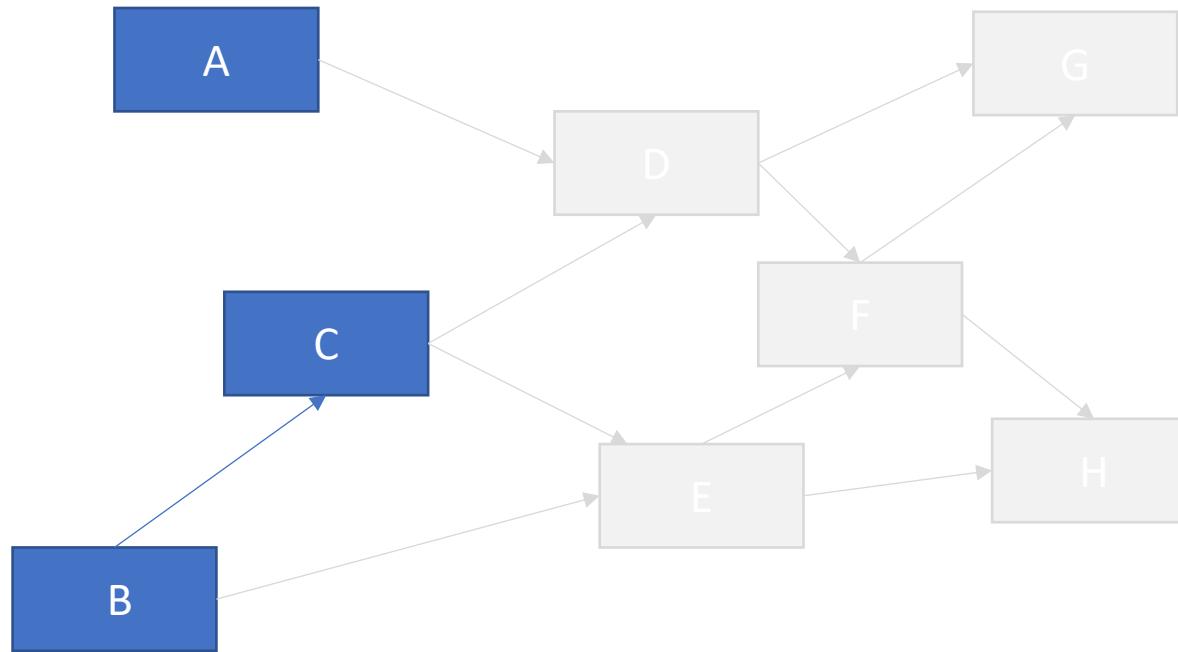
Outside In Migration

Users



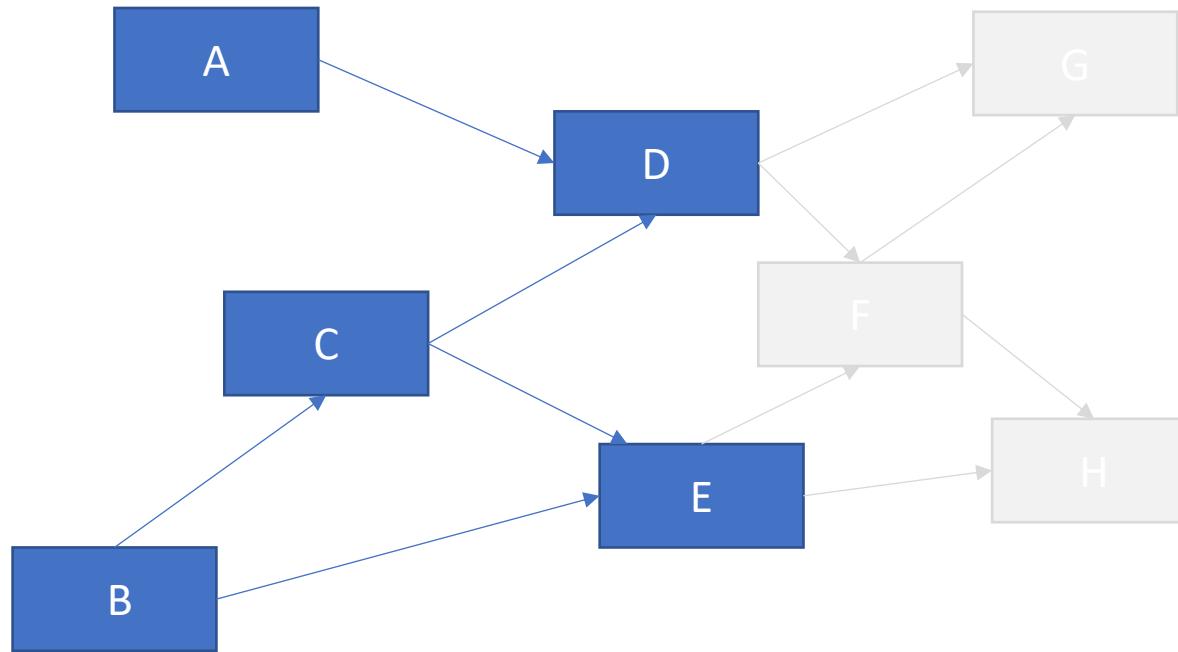
Outside In Migration

Users



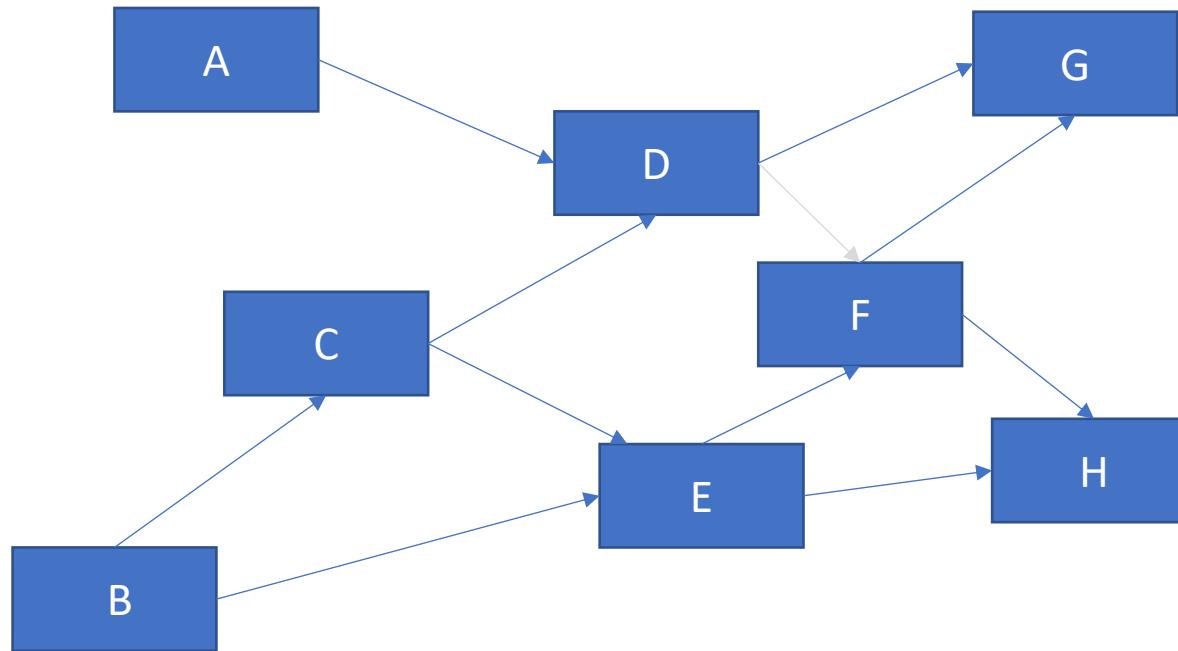
Outside In Migration

Users

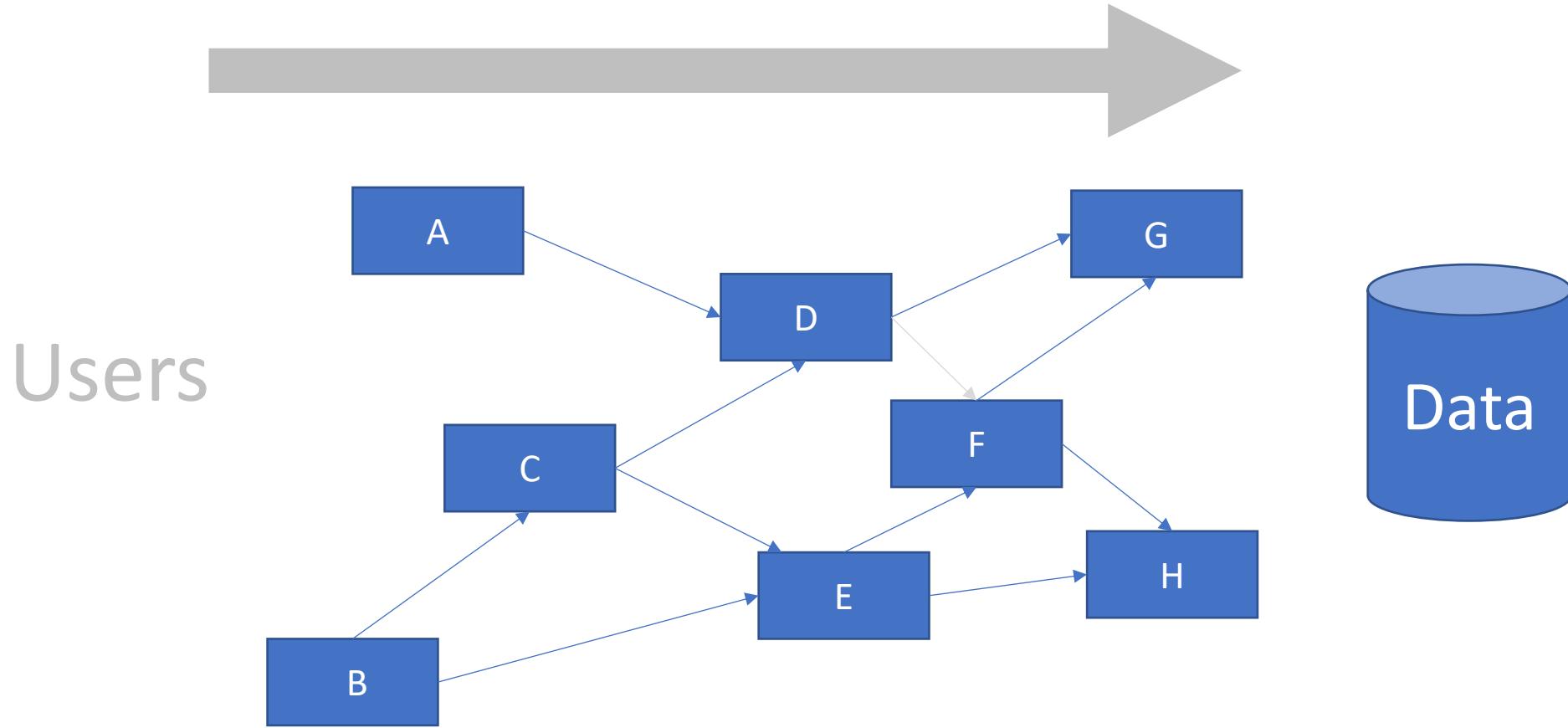


Outside In Migration

Users



Outside In Migration



Other Prioritization Criteria

- Core Services First
- Data Migration
- Cloud Ready
- Measured Performance
- Question: Do you have what you need to prioritize your component migrations?

Cloud Ready Analysis

Cloud Ready Analysis

- Will it run in the cloud (virtual servers)?
- Maintain state within the app itself?
- Custom network tuning?
- Load balancing?

Cloud Ready Analysis

- Will it run in the cloud (virtual servers)?
- Maintain state within the app itself?
- Custom network tuning?
- Load balancing?
- Read: Twelve Factor Application
<https://12factor.net>

Cloud Ready Analysis

- Will it run in the cloud (virtual servers)?
- Maintain state within the app itself?
- Custom network tuning?
- Load balancing?
- Read: Twelve Factor Application
<https://12factor.net>
- Application Performance Monitoring (APM)

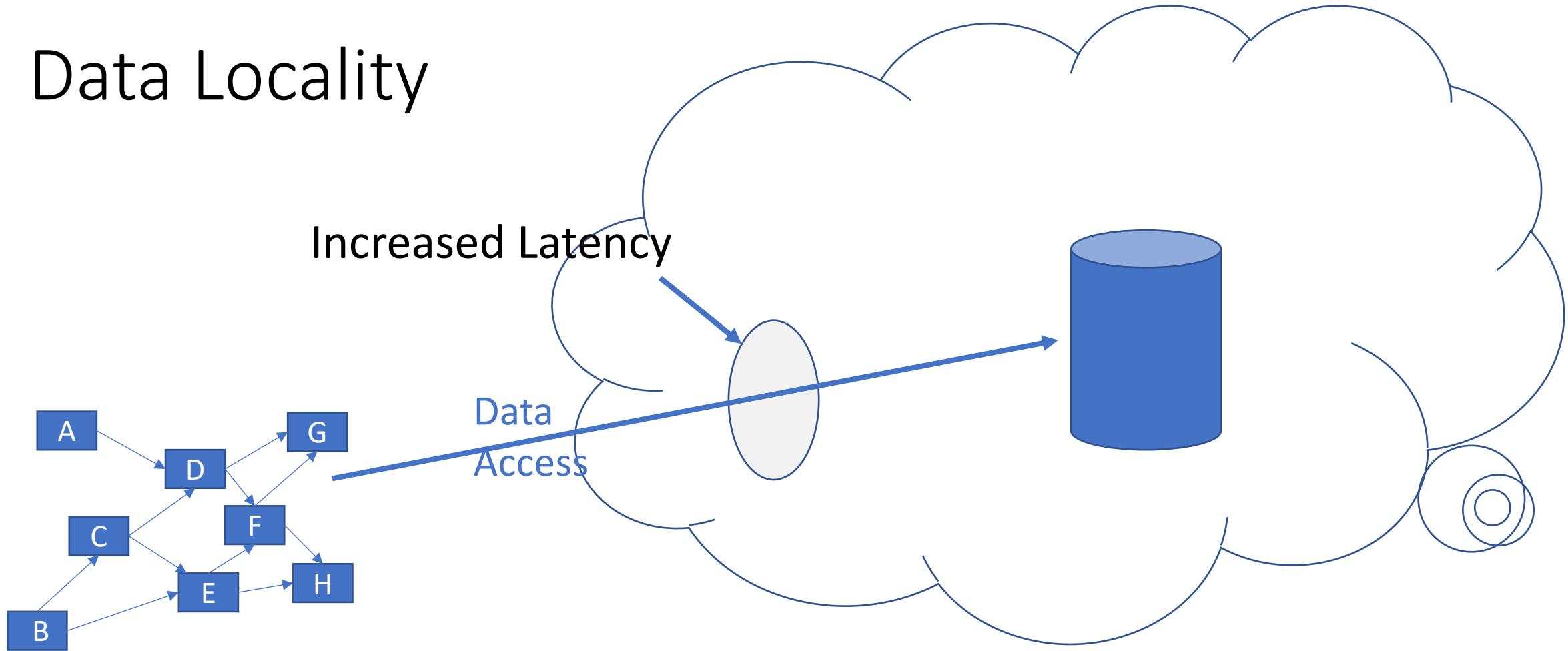
Planned Refactoring

Planned Refactoring

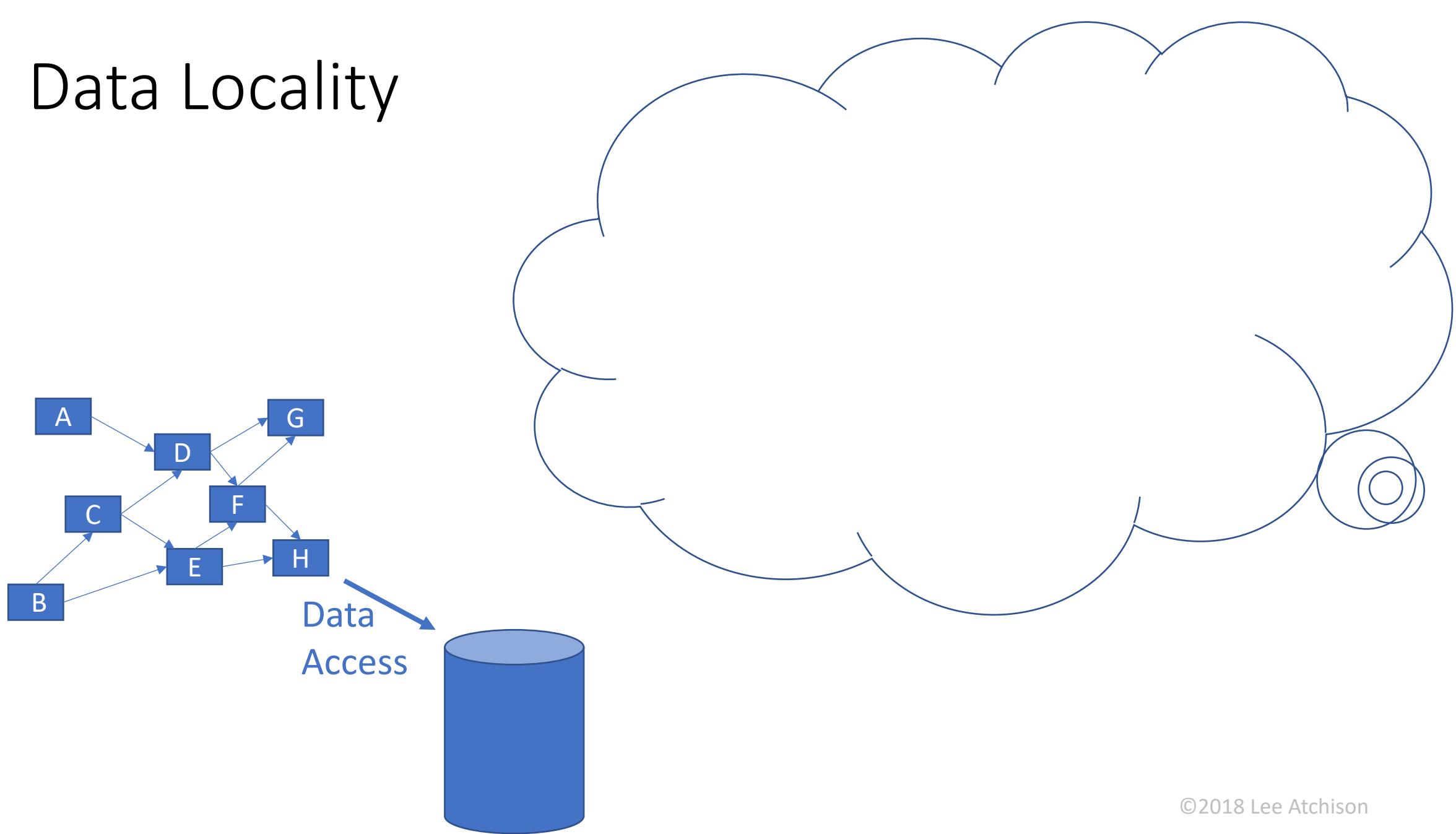
- Support variable number of instances
- Dynamic cloud resources
- Make migration easier
 - Access data store
 - Ease data migration
 - Minimize downtime
- Monolith → Service Architecture

Data Migration

Data Locality

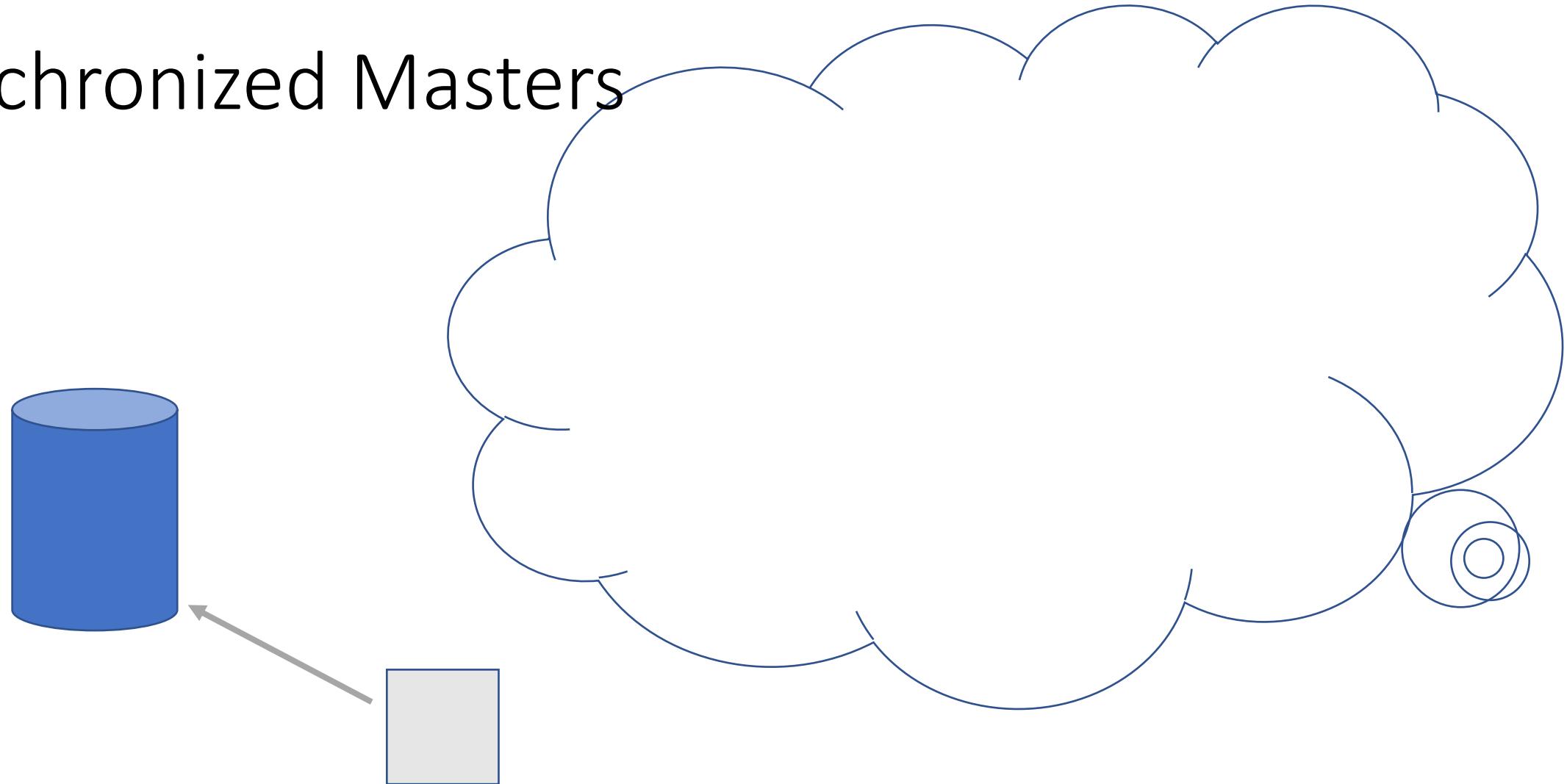


Data Locality

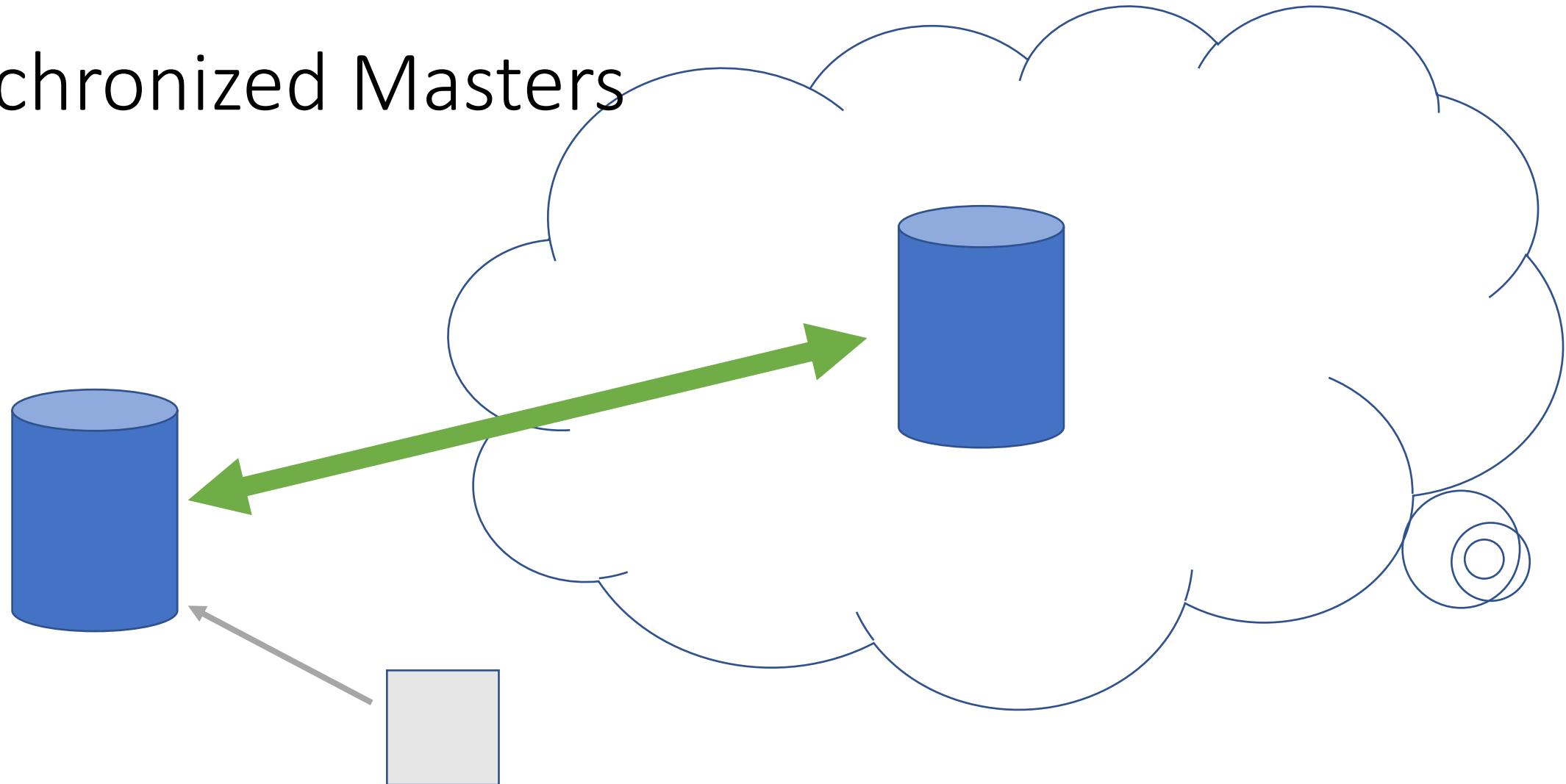


Synchronized Masters

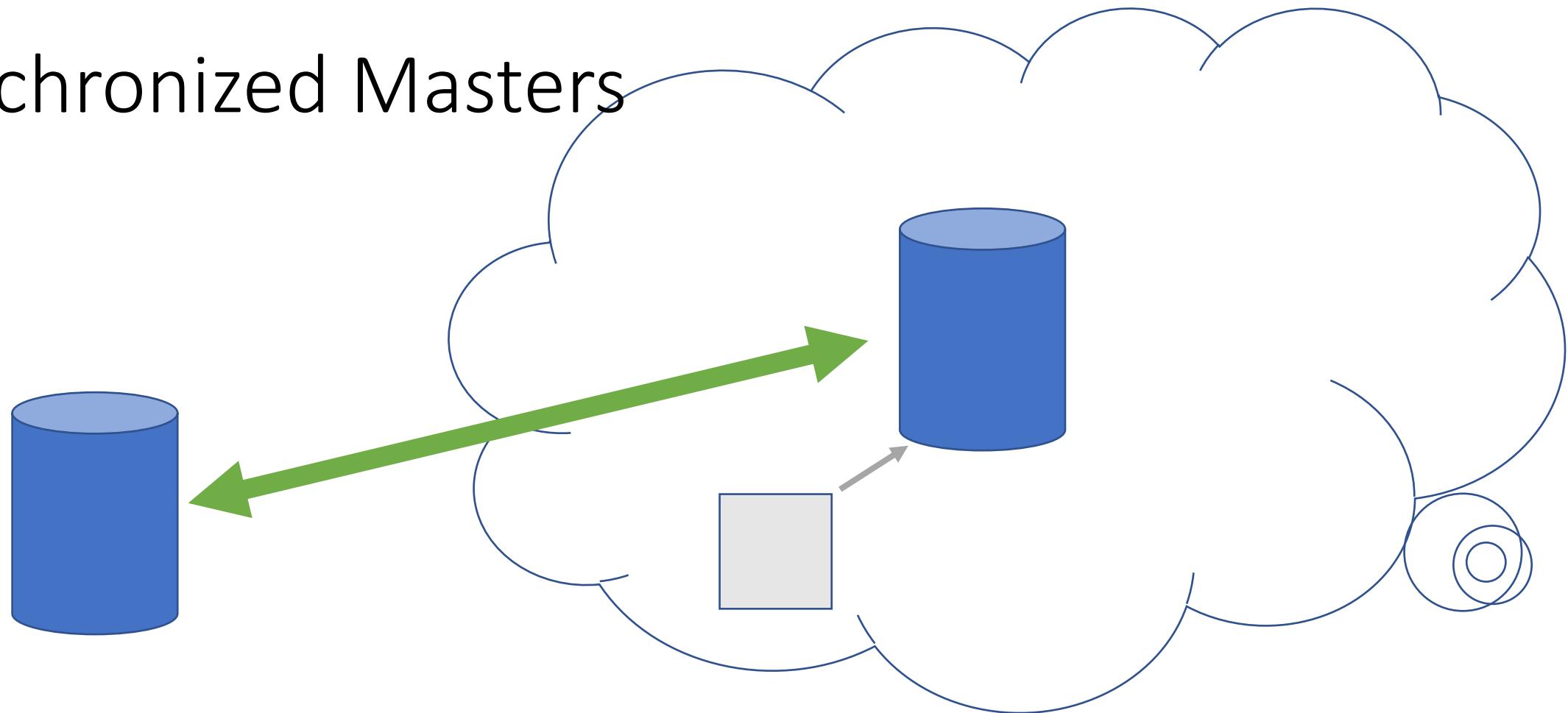
Synchronized Masters



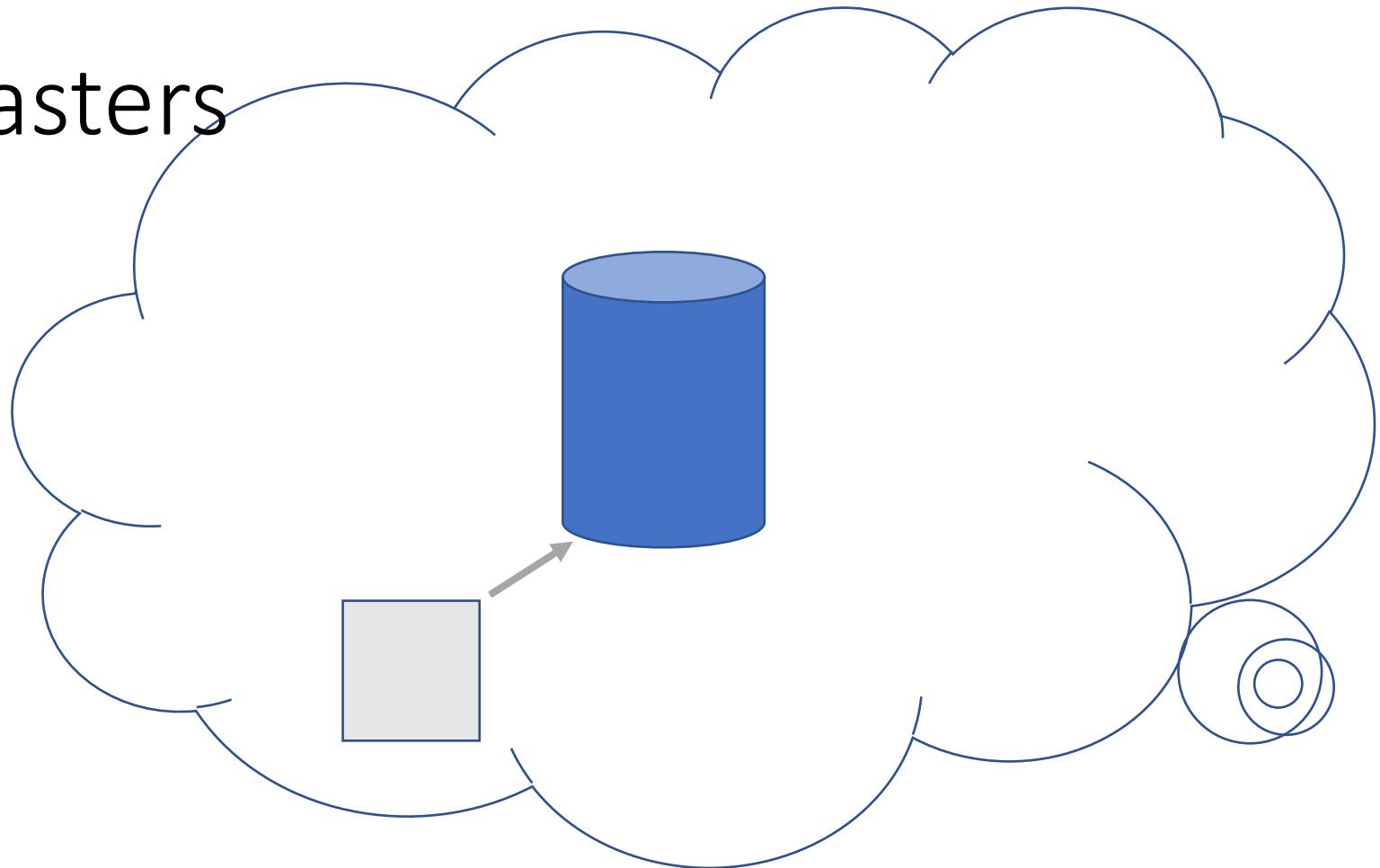
Synchronized Masters



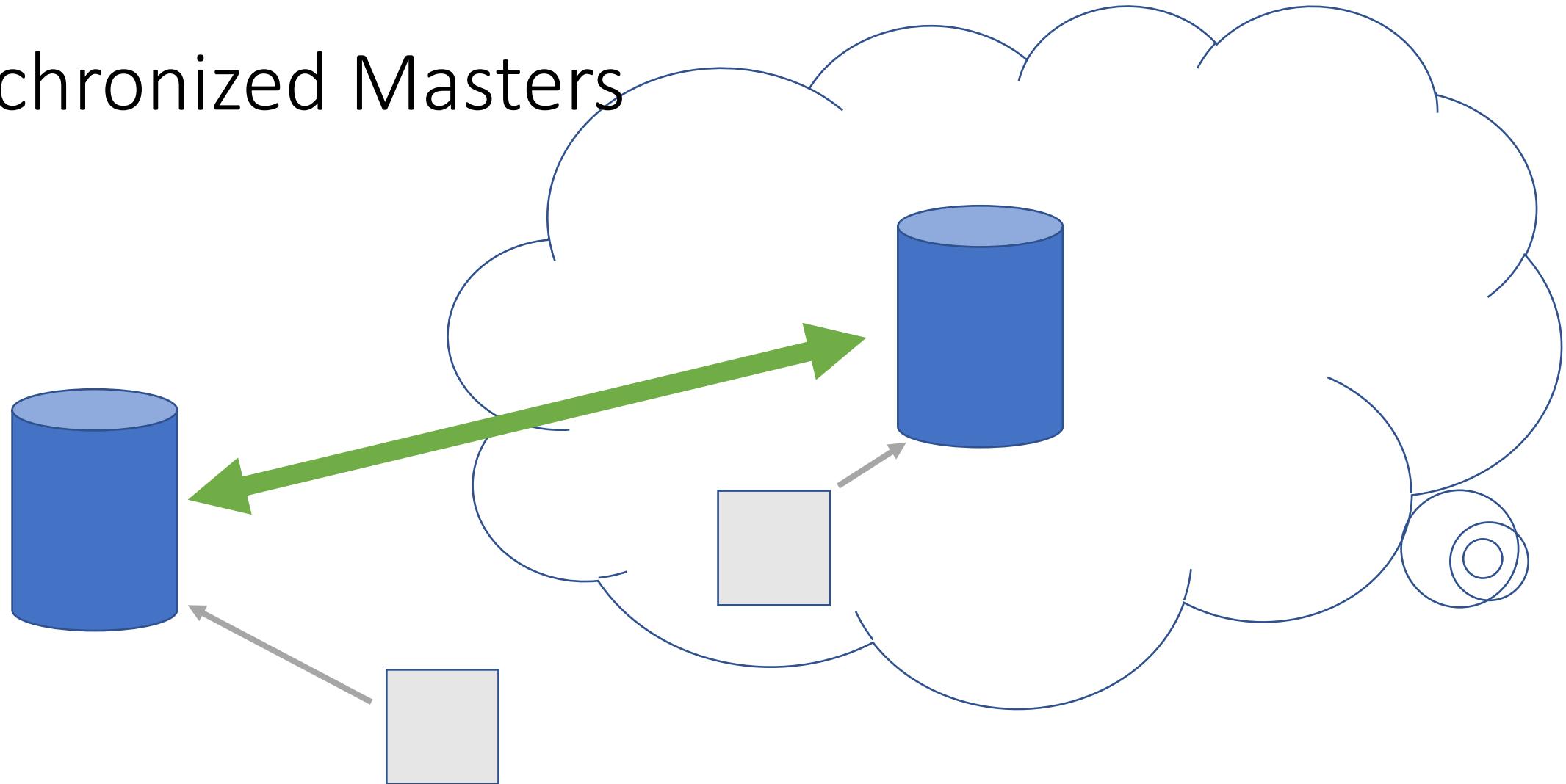
Synchronized Masters



Synchronized Masters

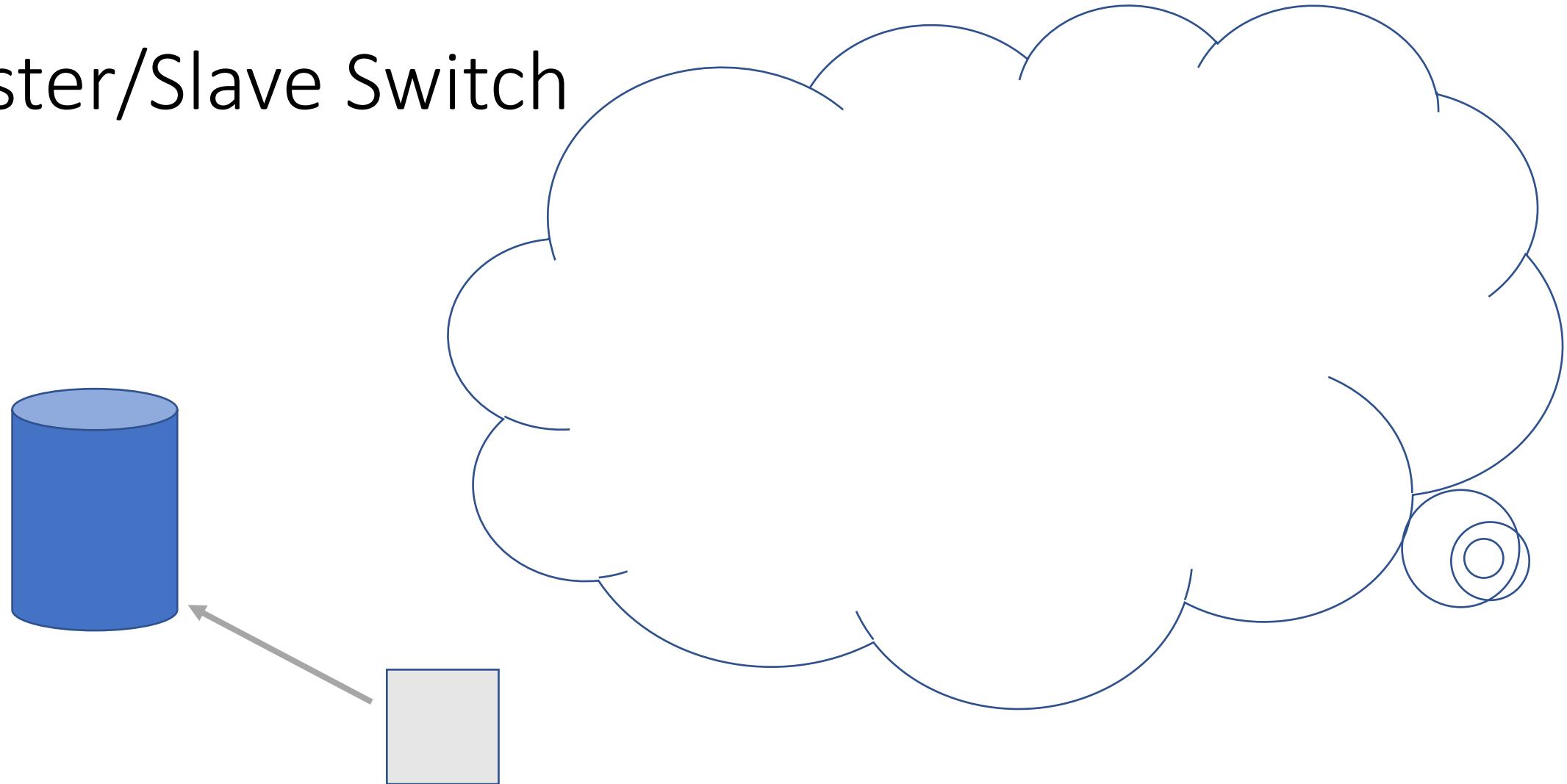


Synchronized Masters

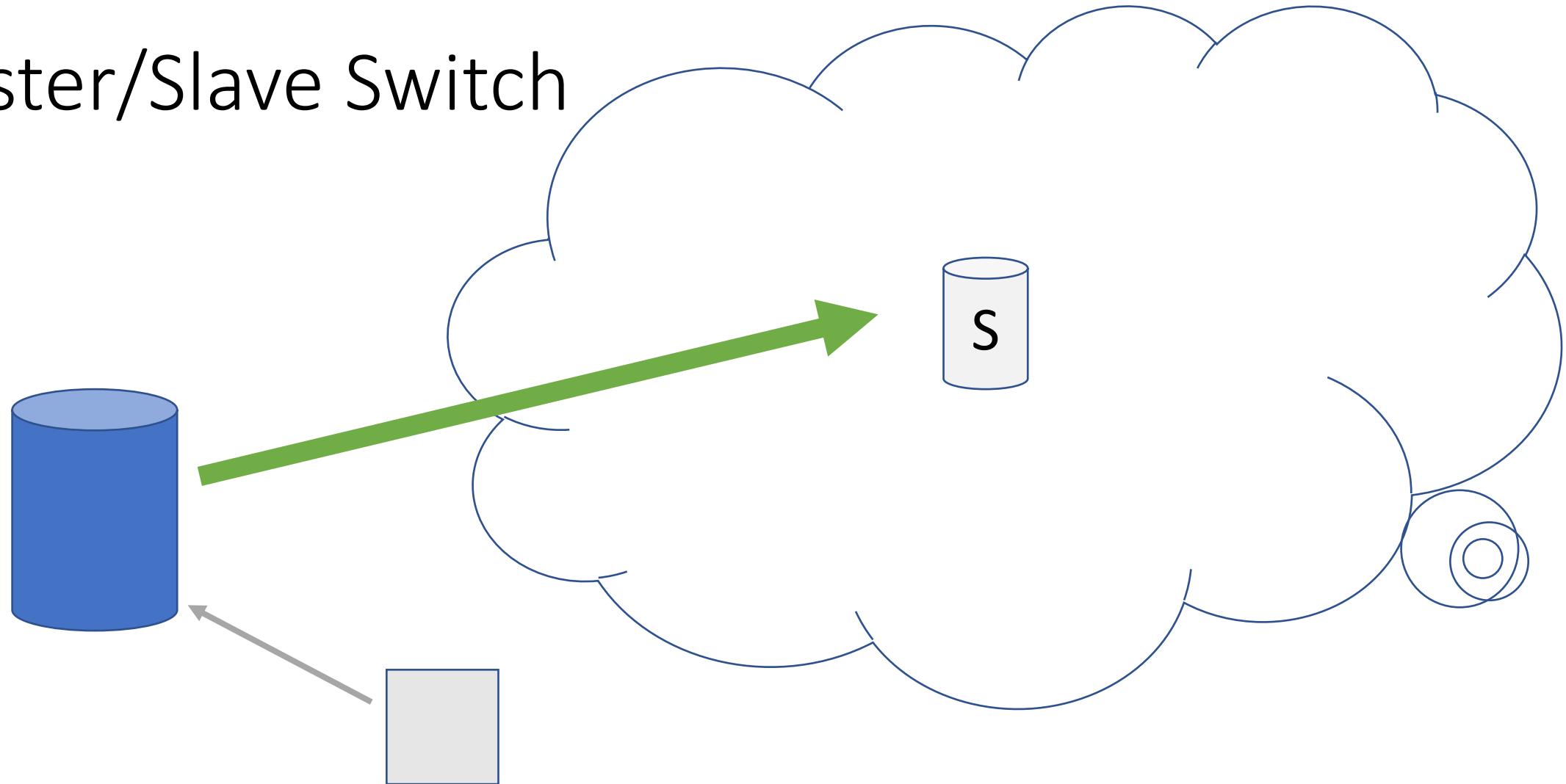


Master/Slave Switch

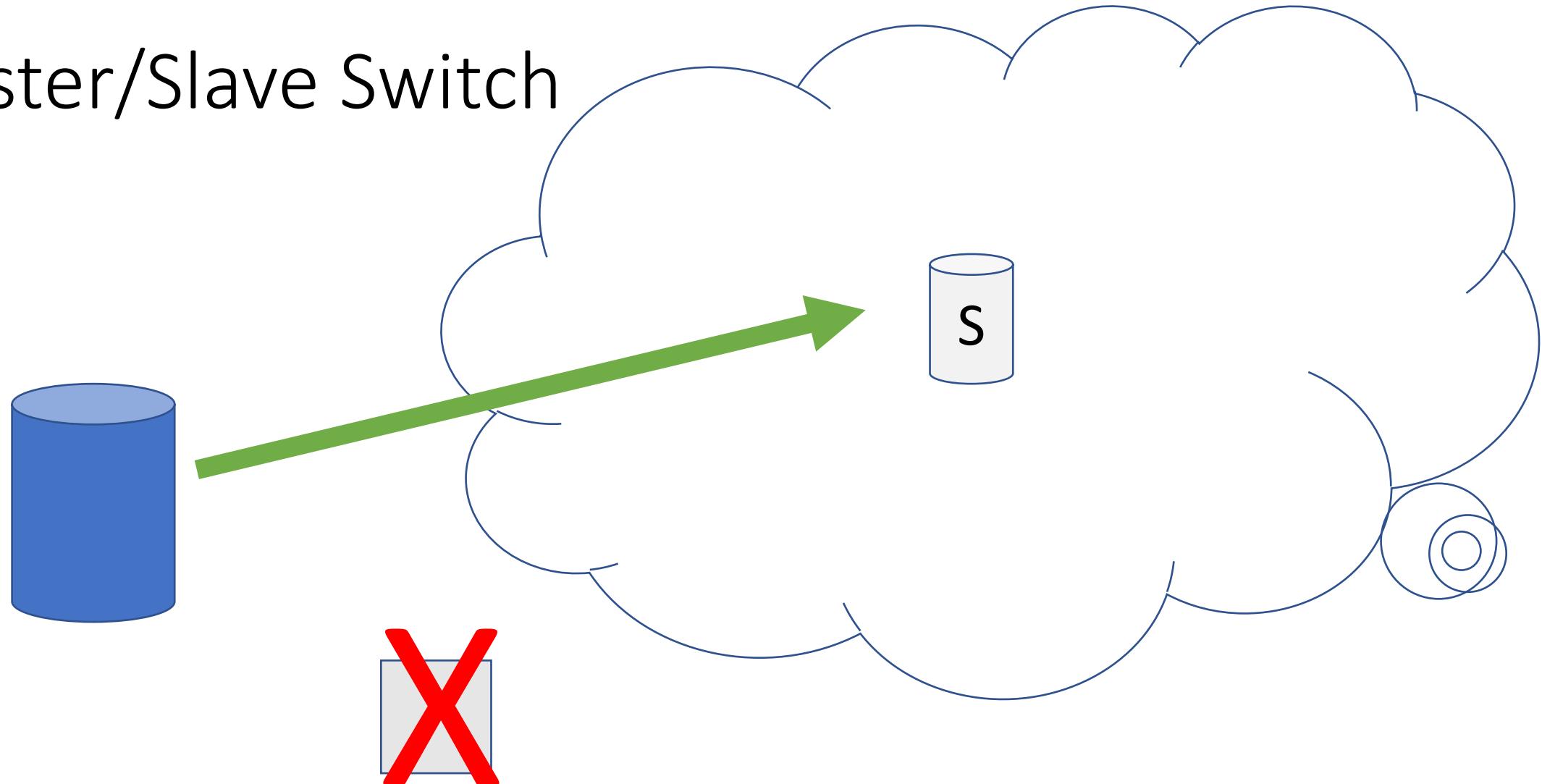
Master/Slave Switch



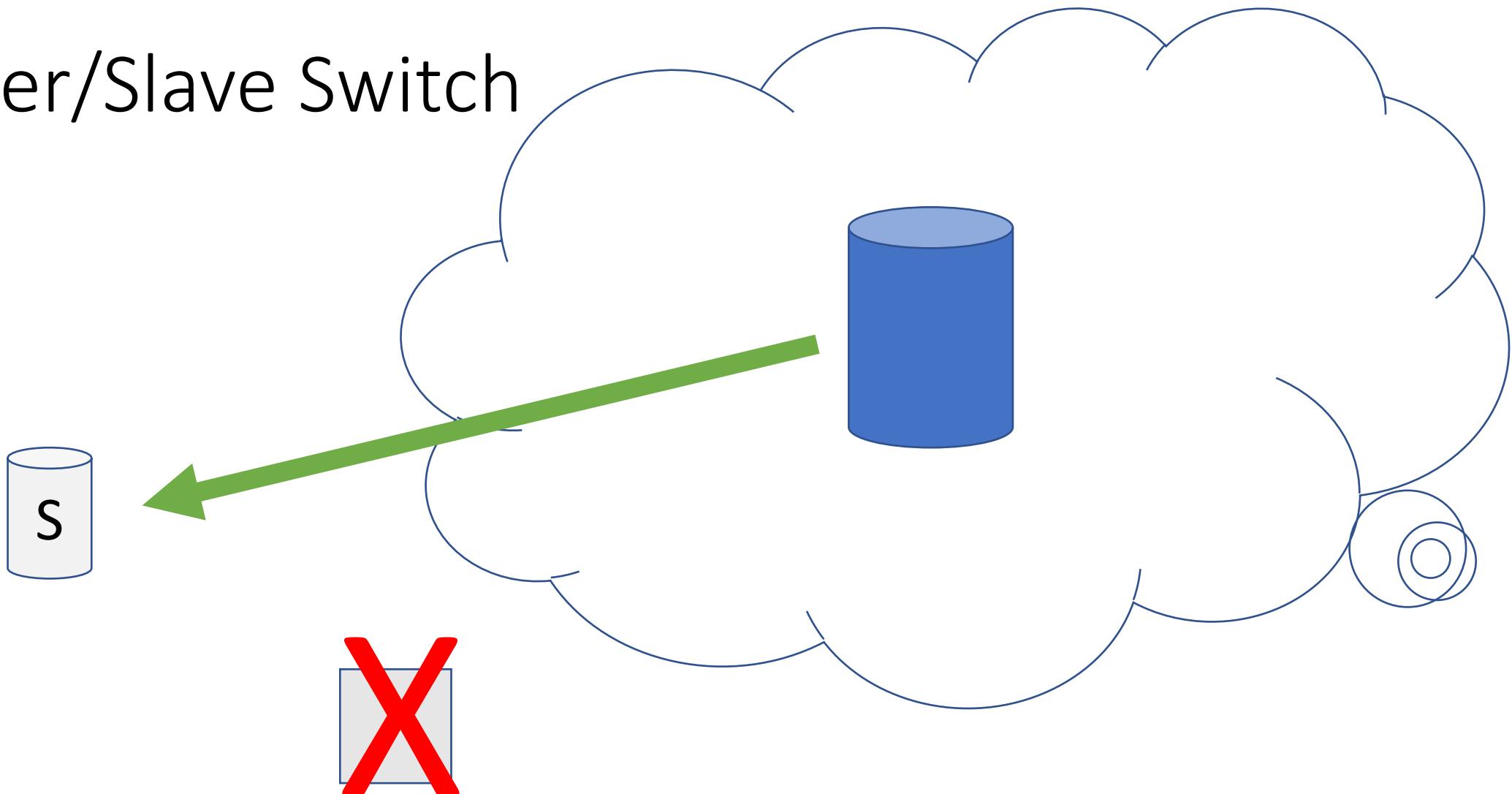
Master/Slave Switch



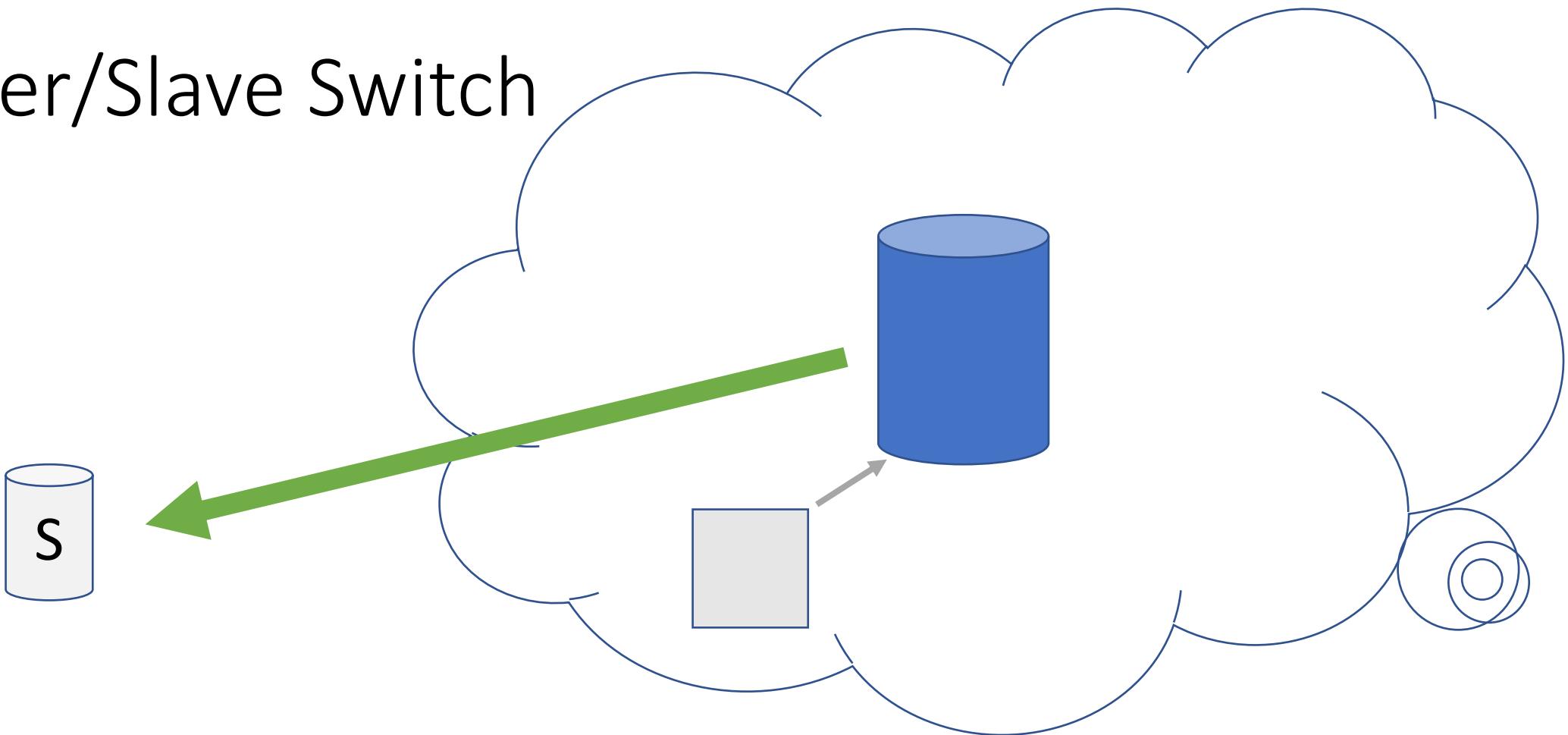
Master/Slave Switch



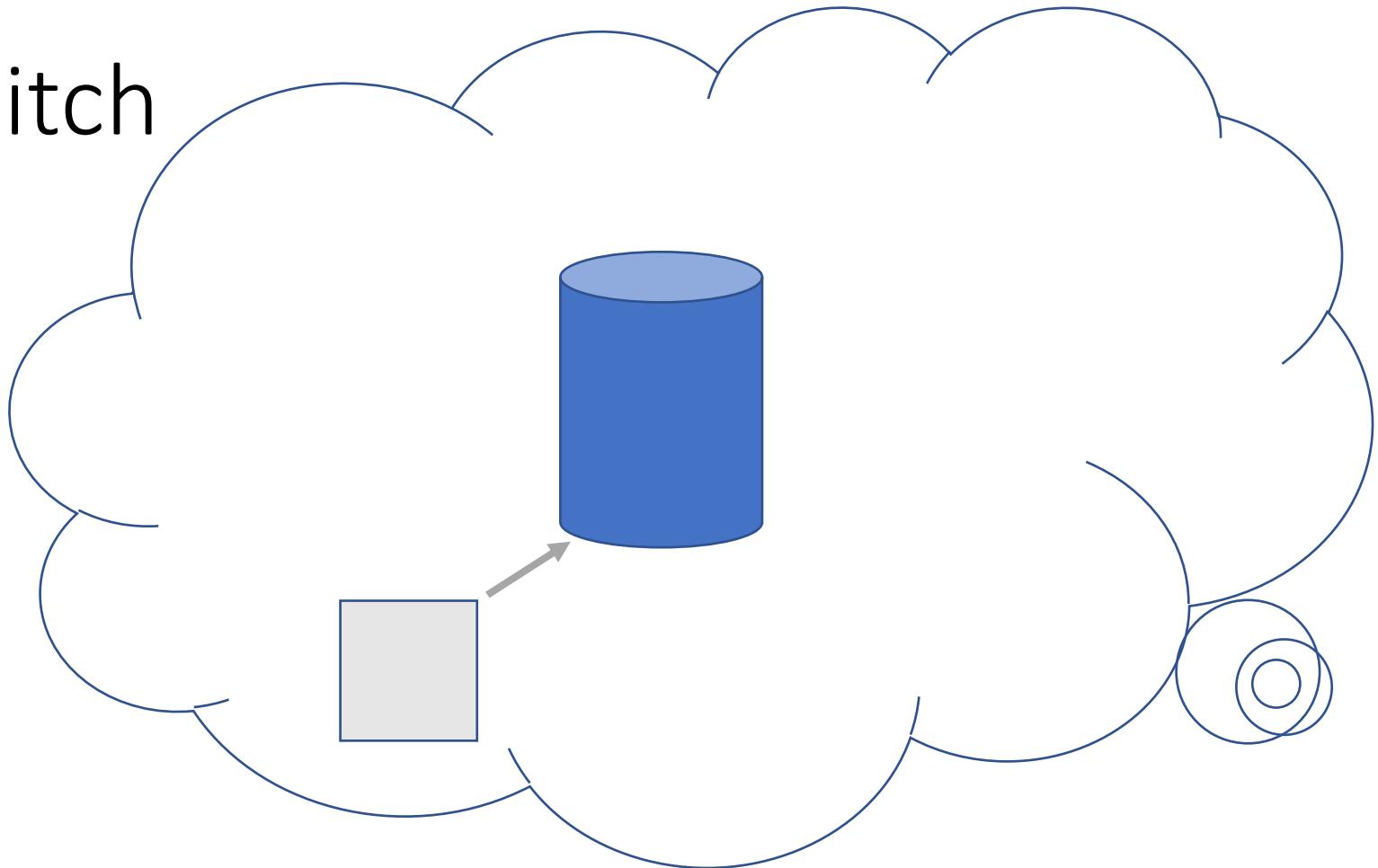
Master/Slave Switch



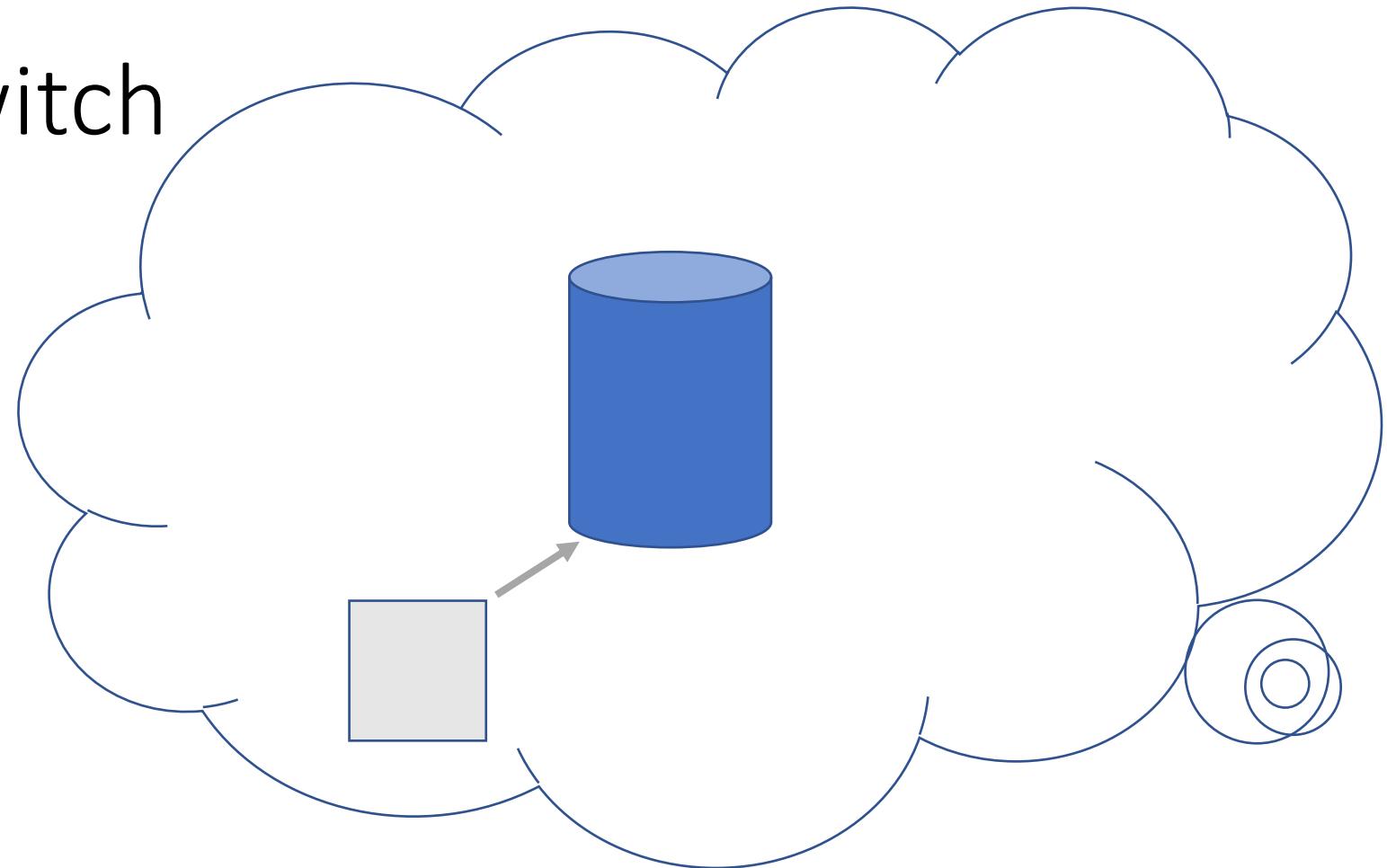
Master/Slave Switch



Master/Slave Switch



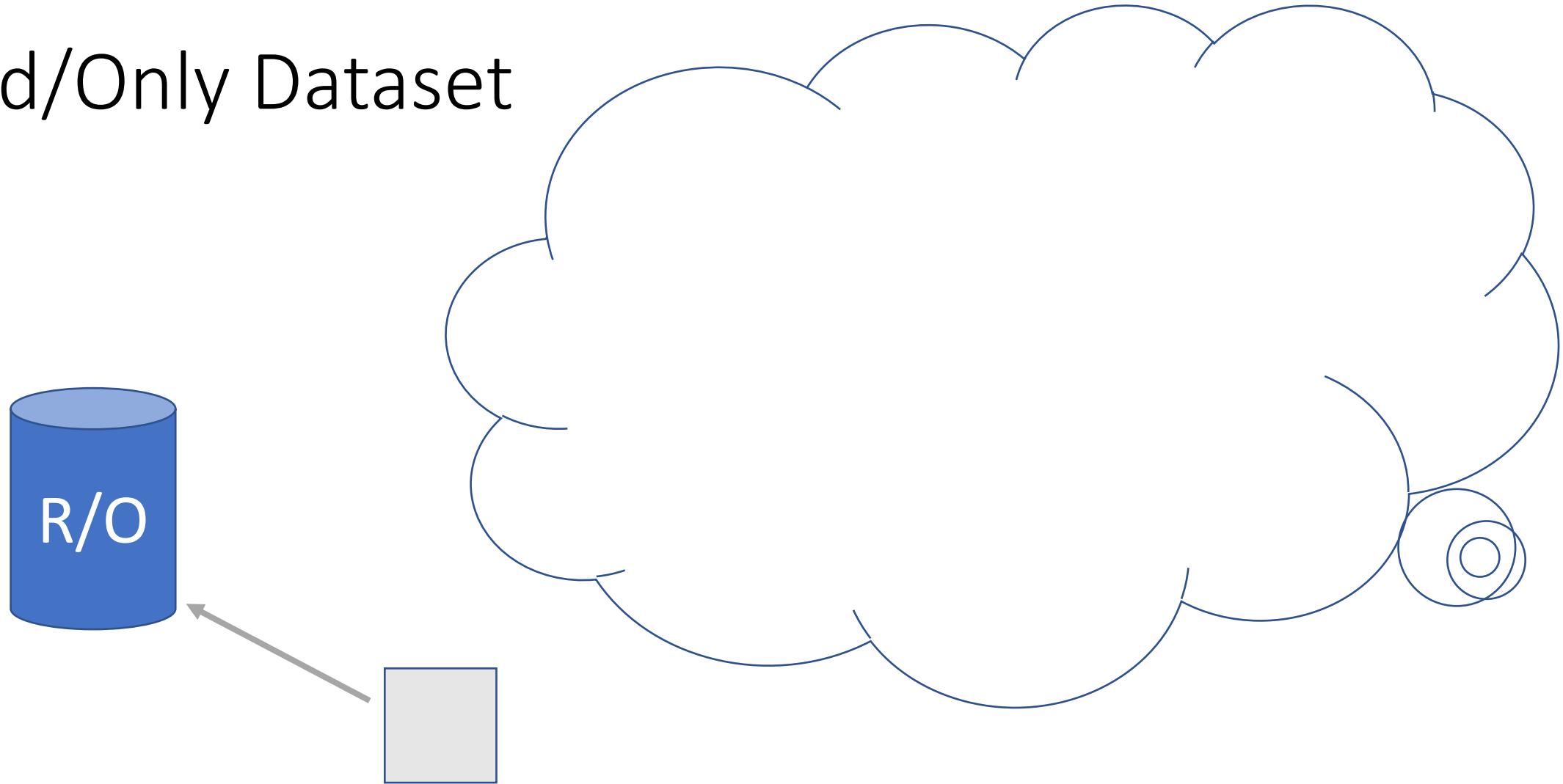
Master/Slave Switch



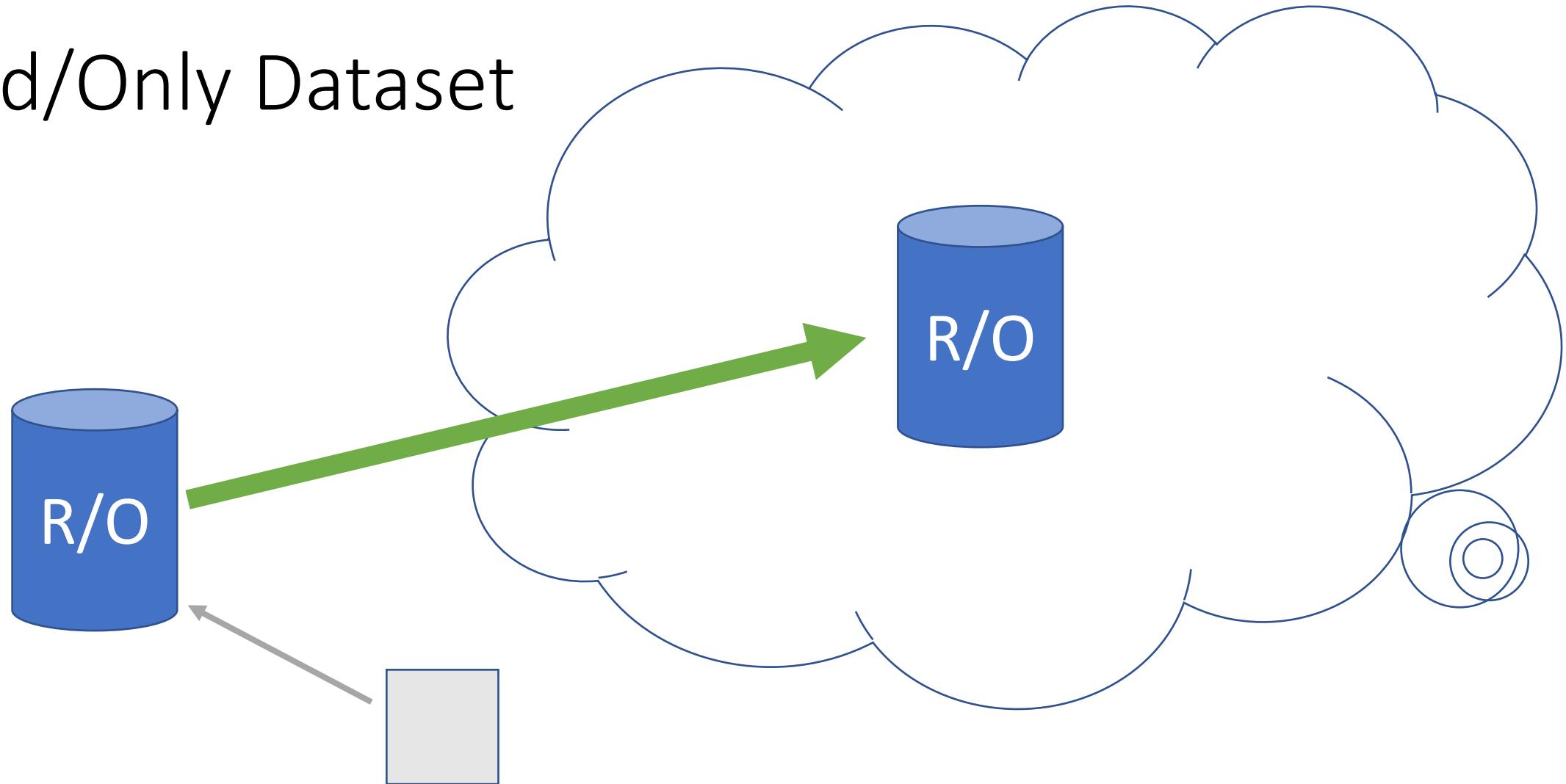
X → Means downtime

Read/Only Dataset

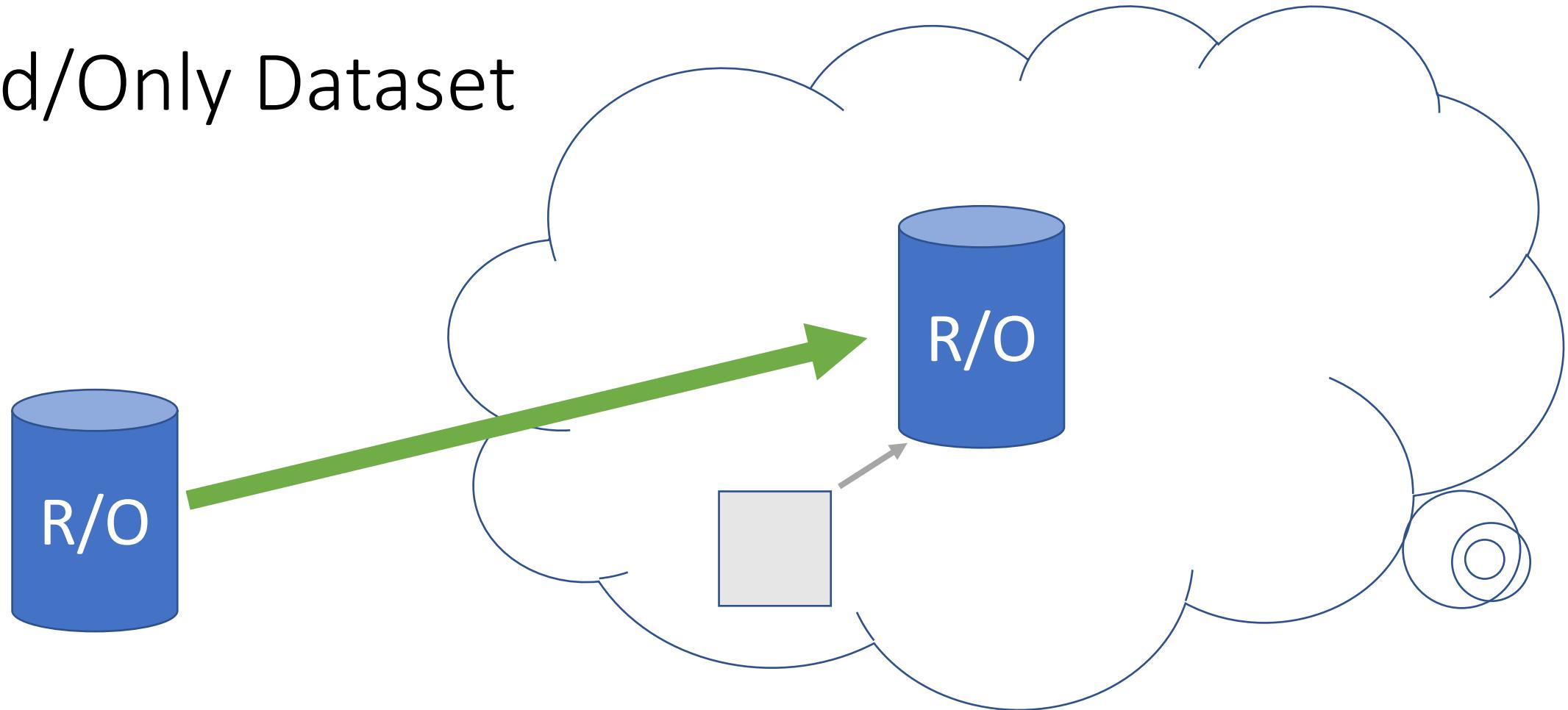
Read/Only Dataset



Read/Only Dataset

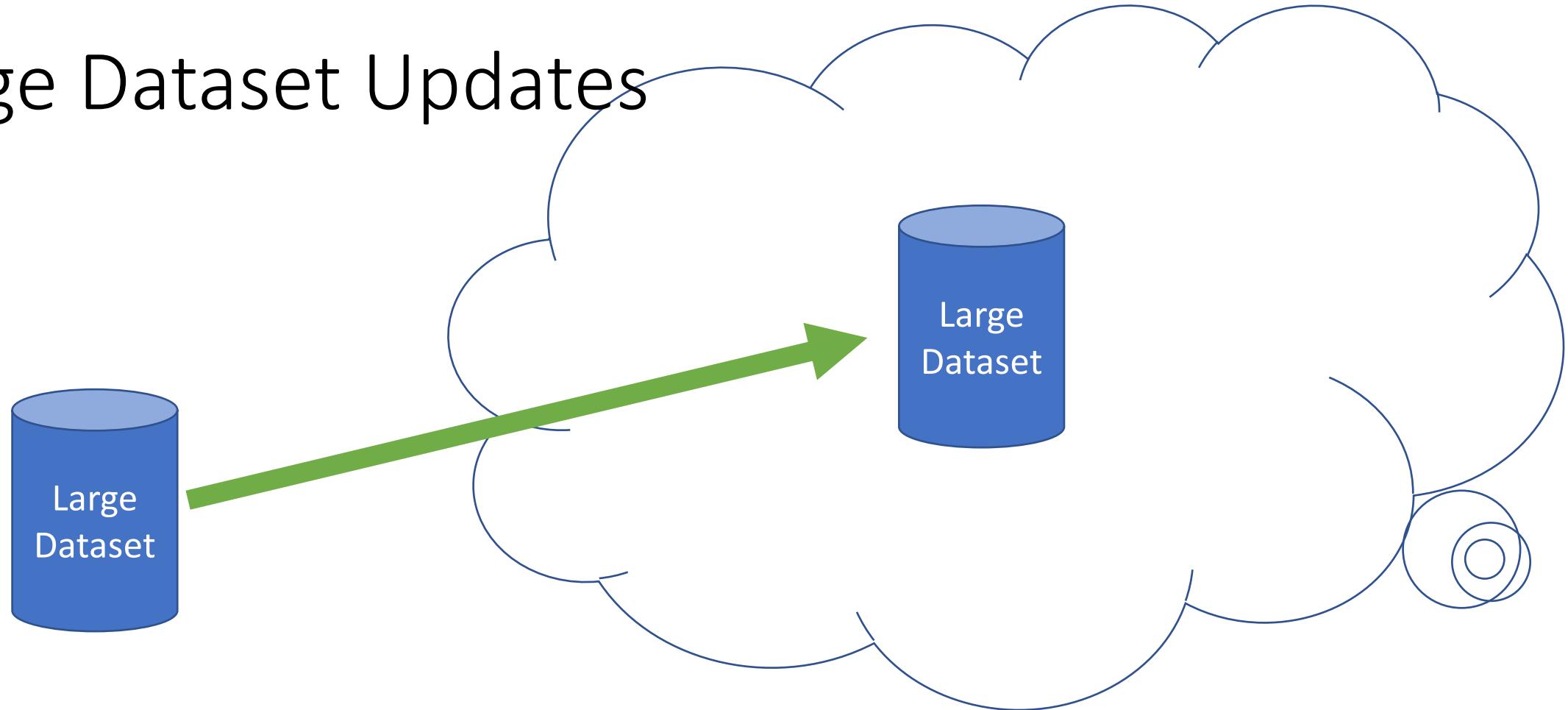


Read/Only Dataset

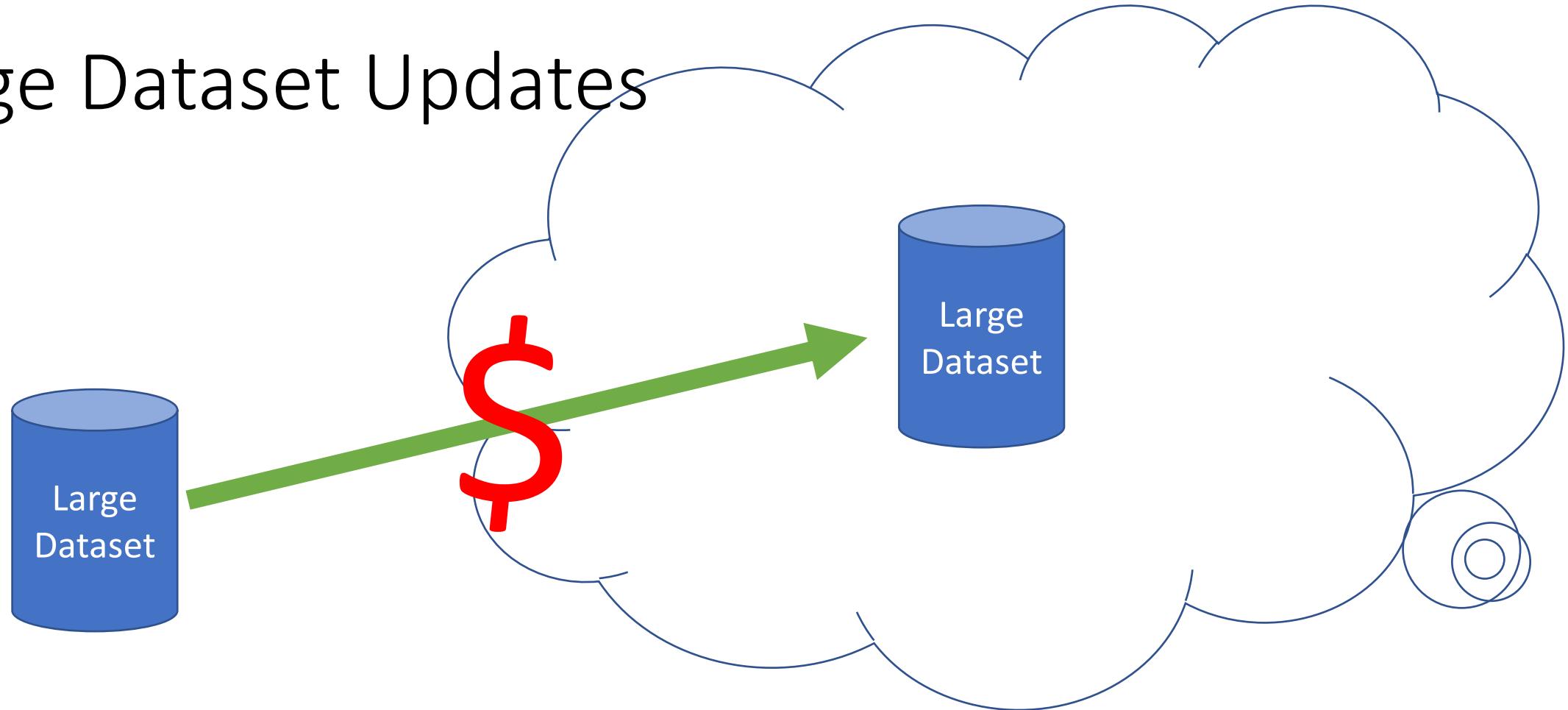


Large Dataset Updates

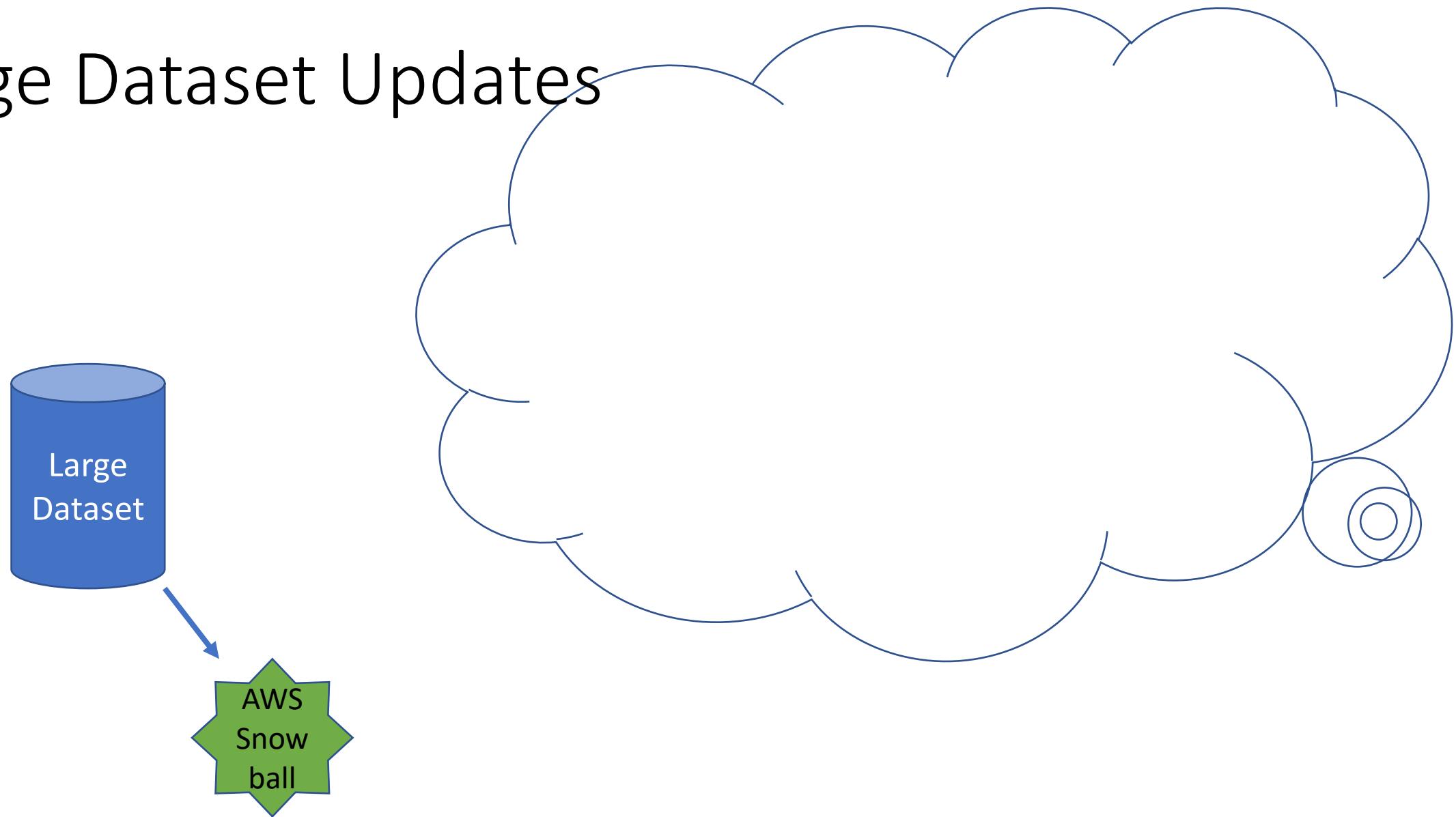
Large Dataset Updates



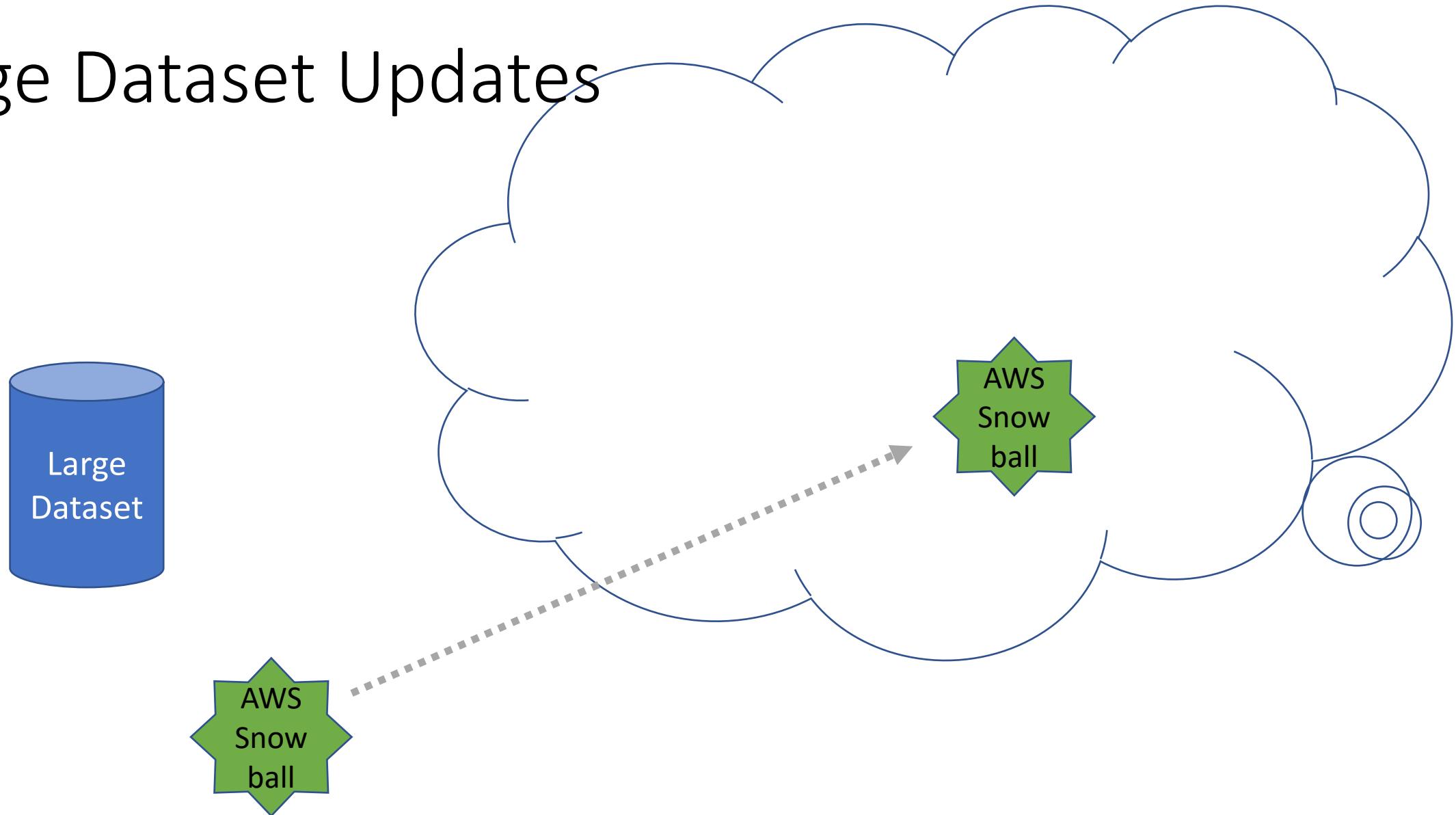
Large Dataset Updates



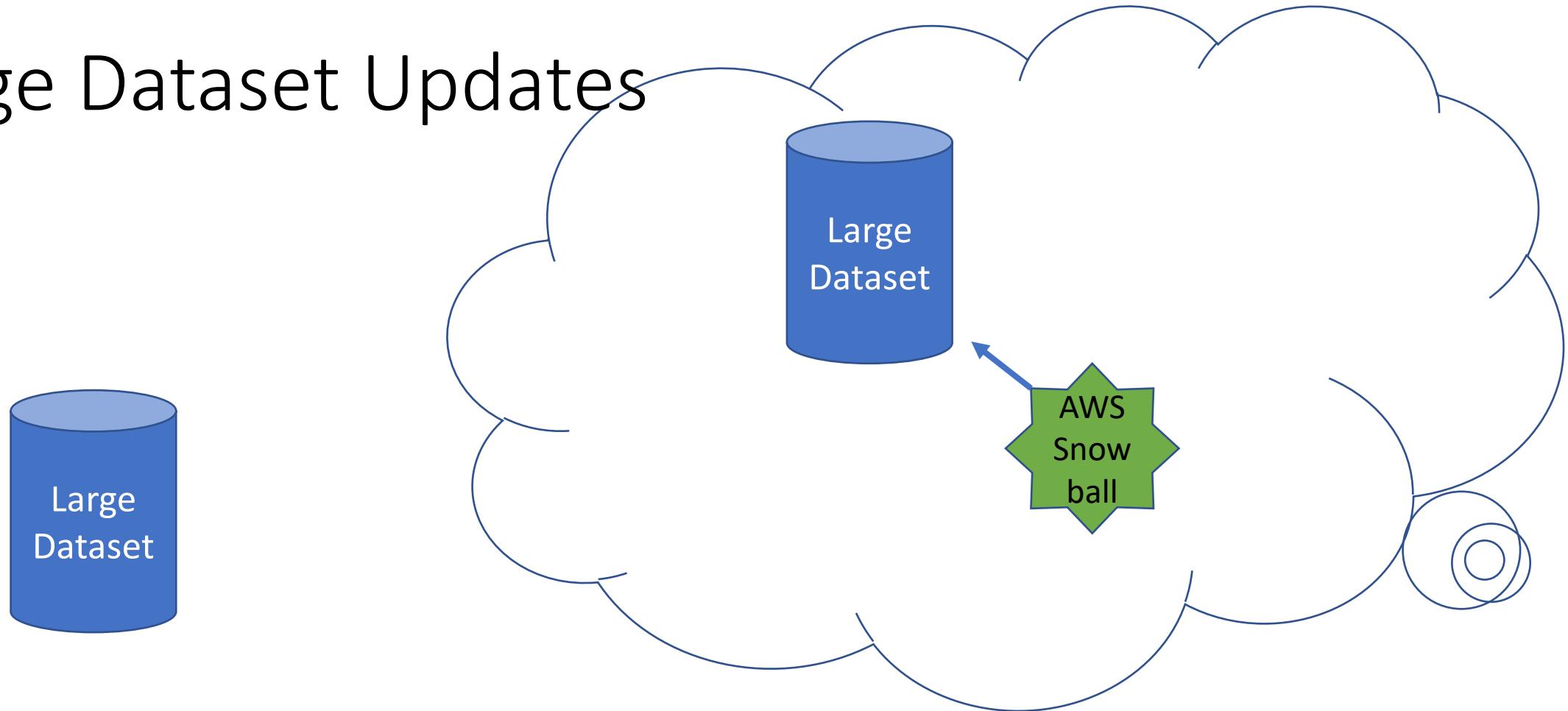
Large Dataset Updates



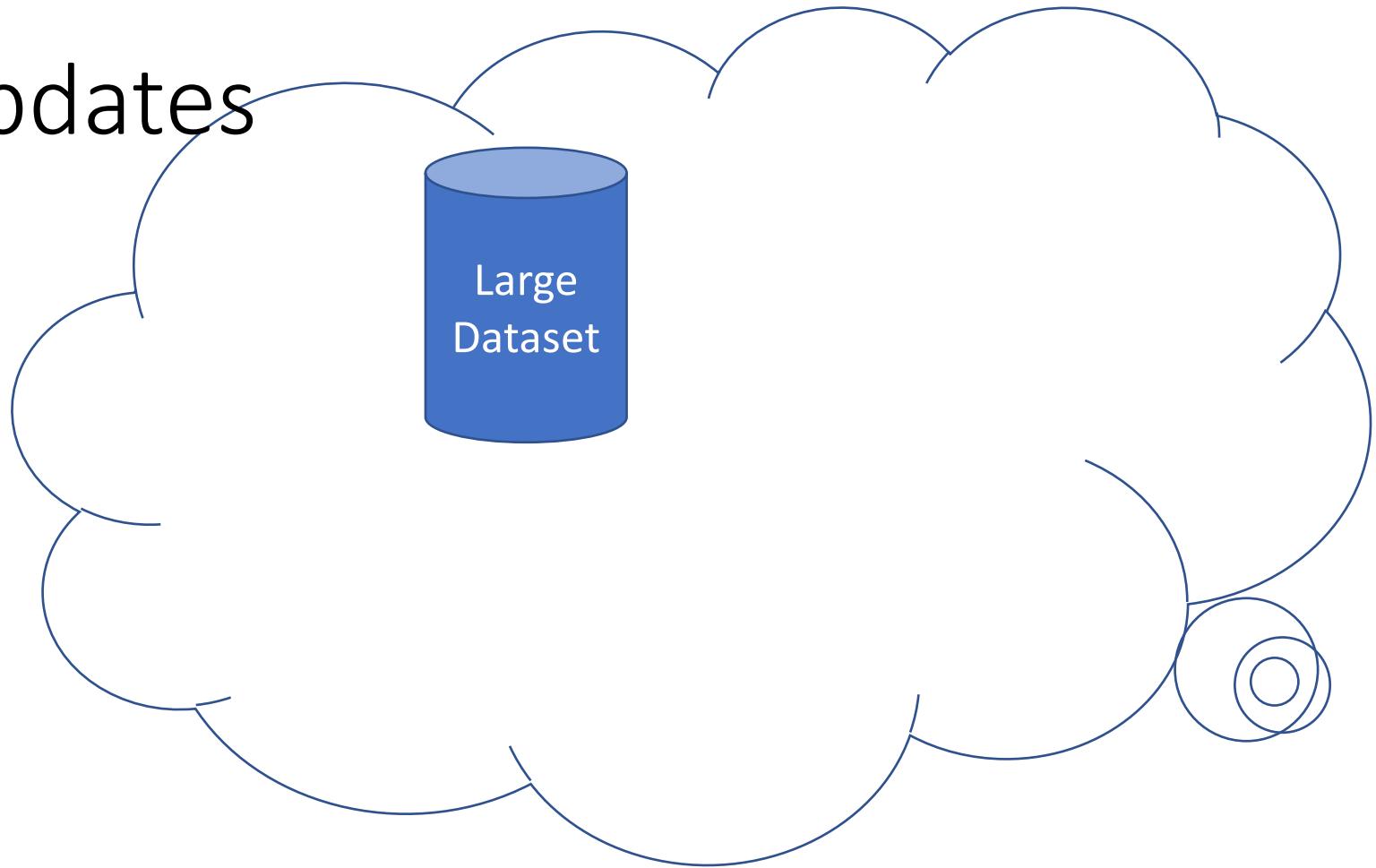
Large Dataset Updates



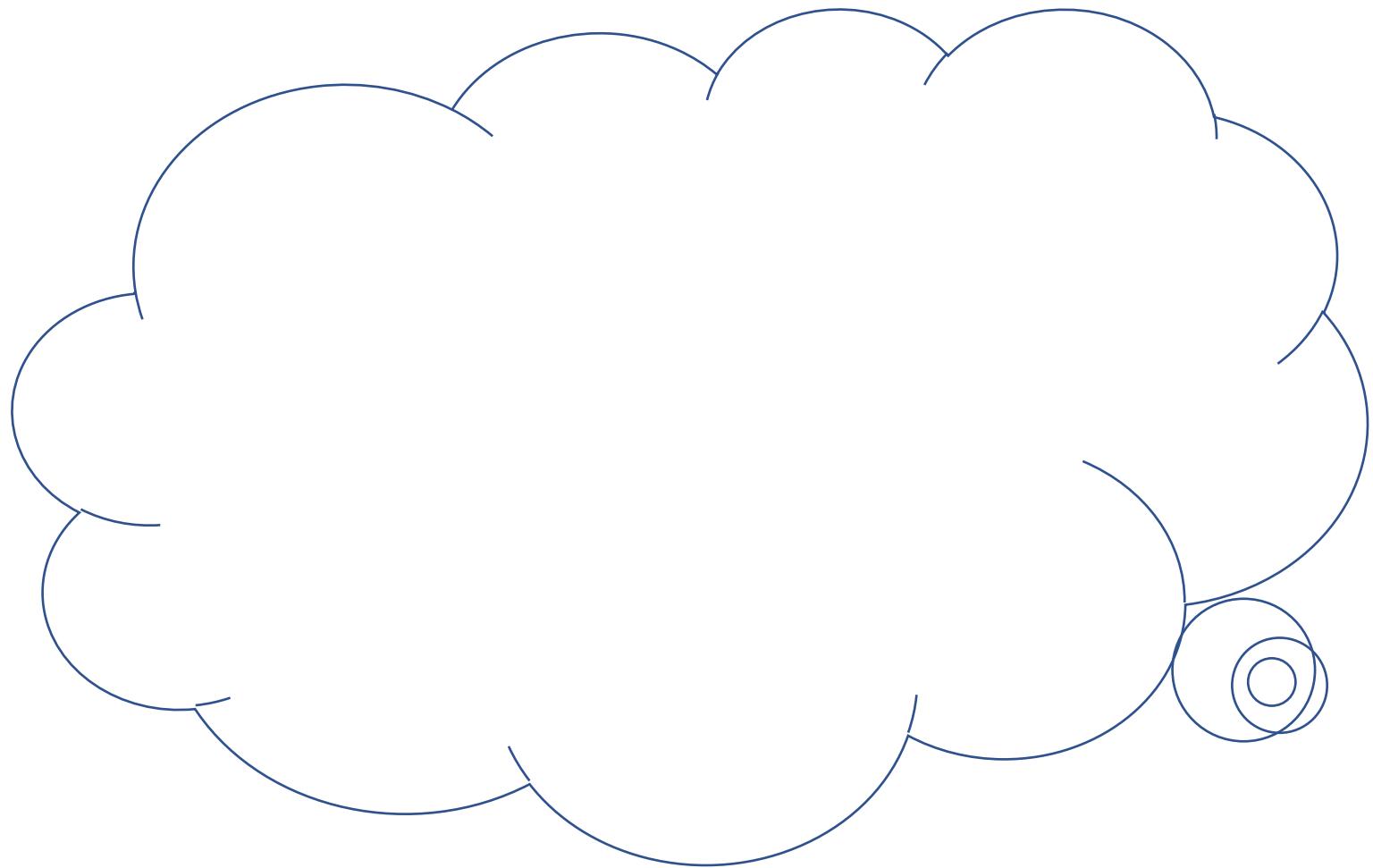
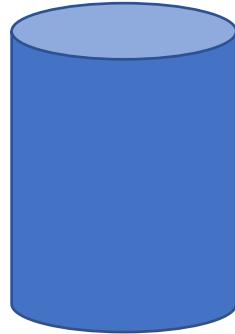
Large Dataset Updates



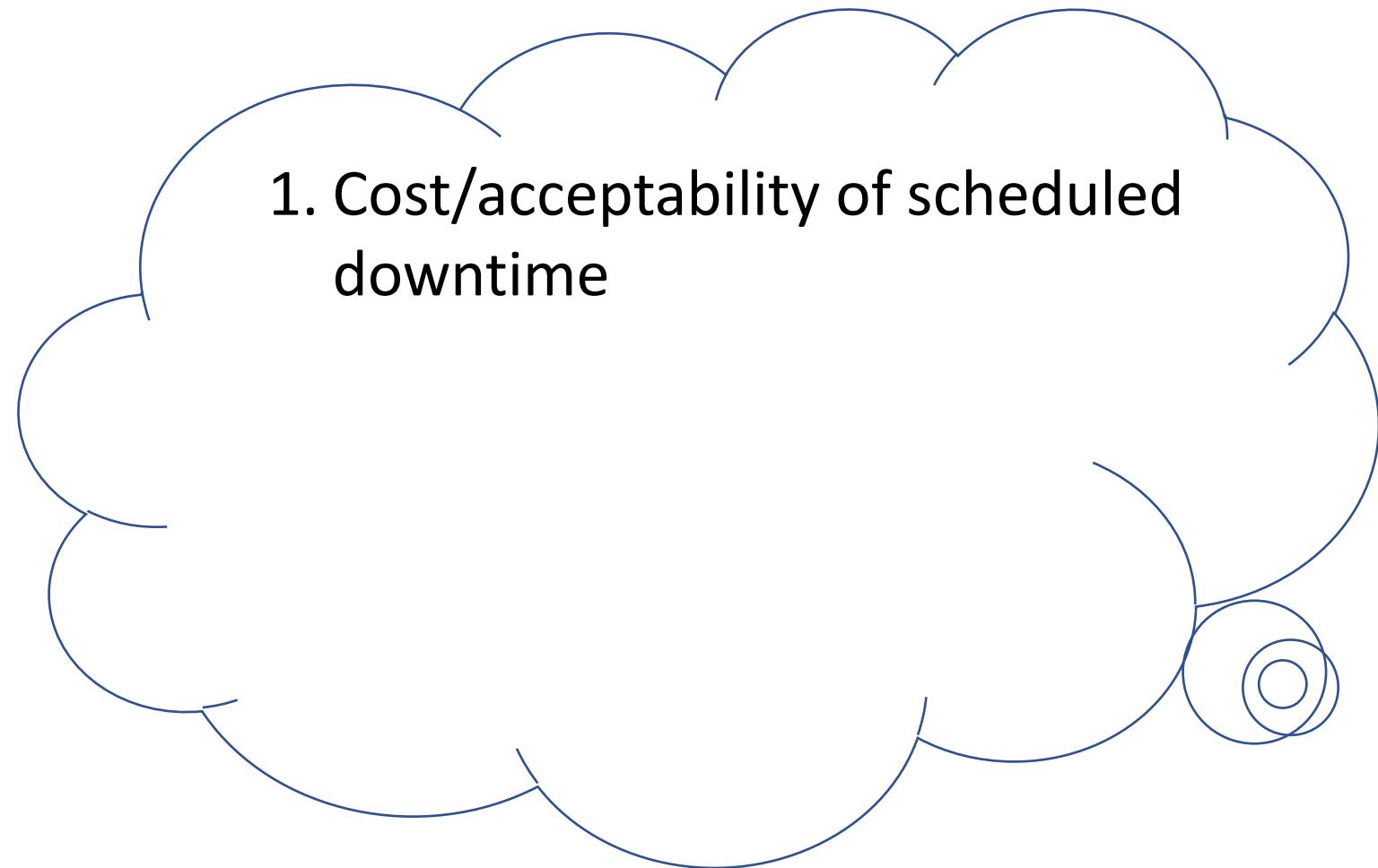
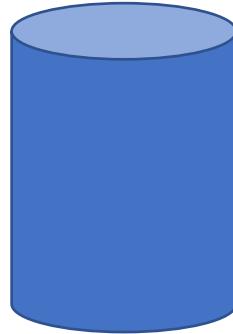
Large Dataset Updates



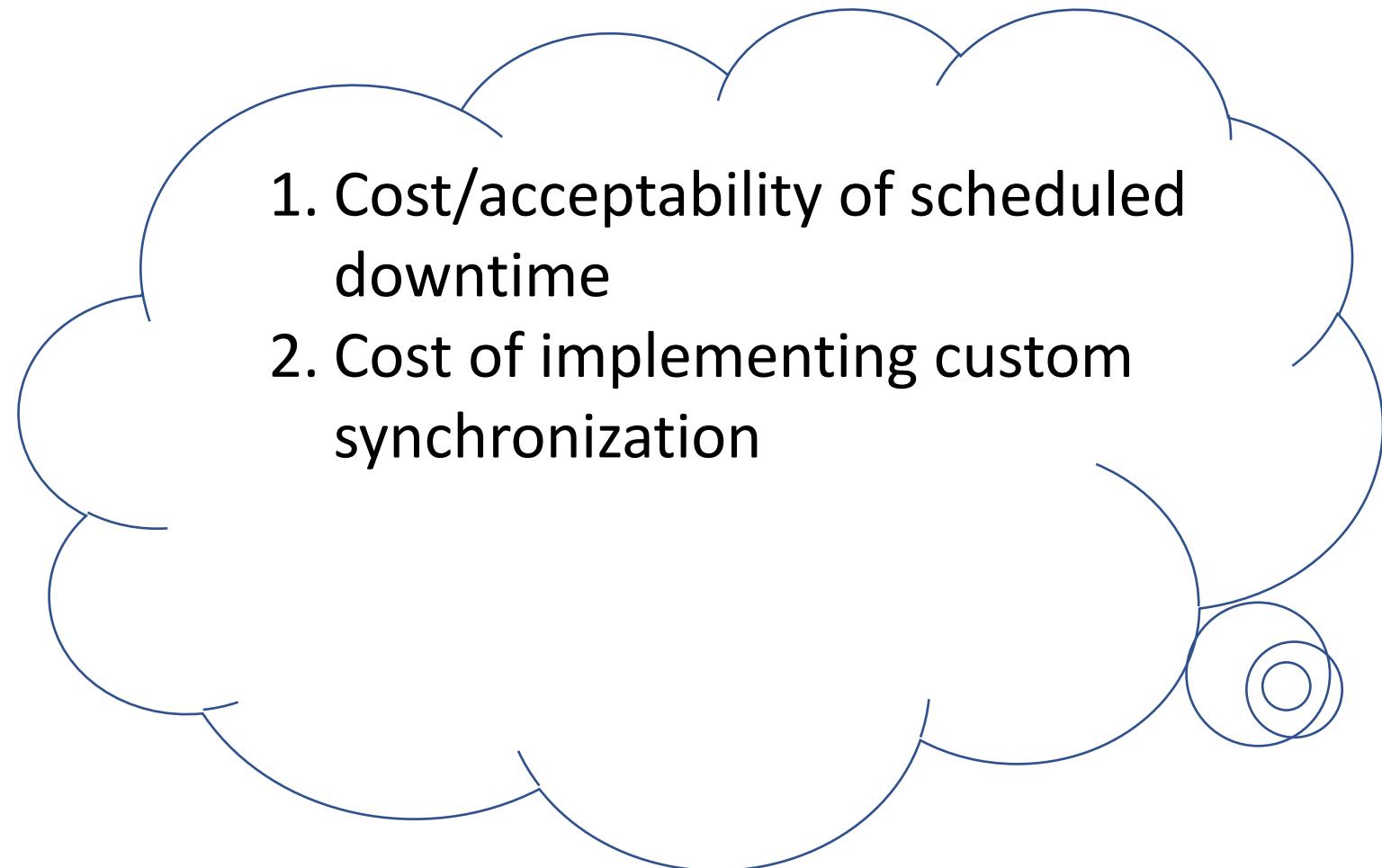
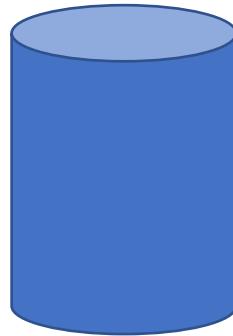
Variations & Combinations



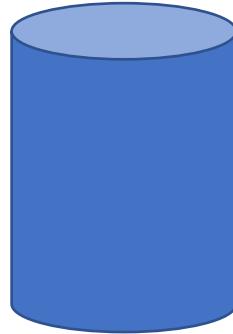
Variations & Combinations



Variations & Combinations

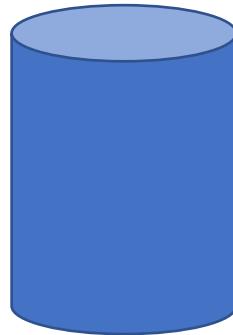


Variations & Combinations



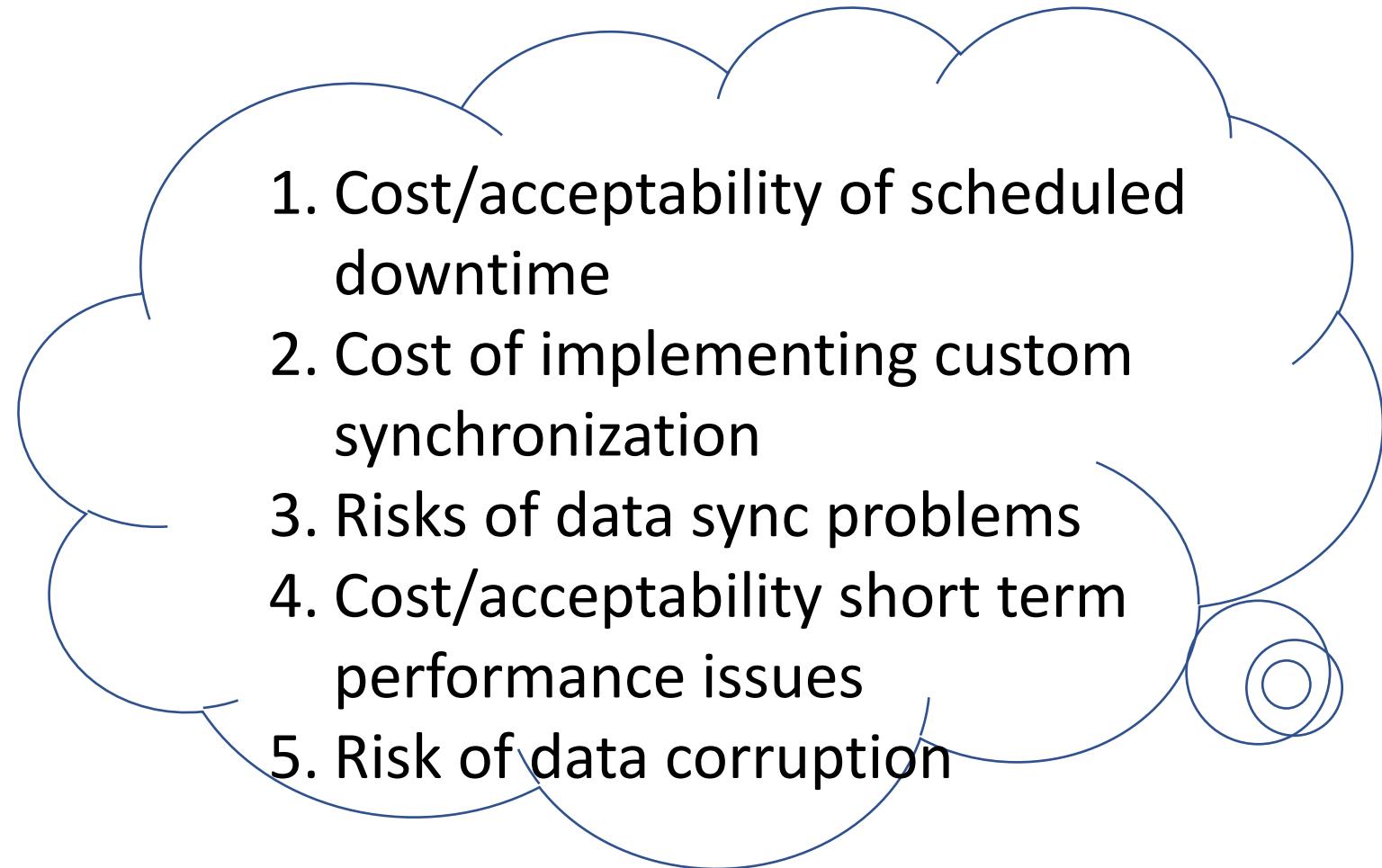
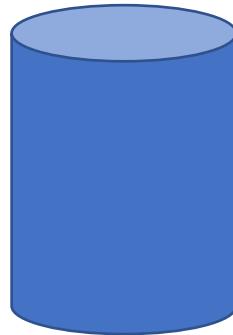
1. Cost/acceptability of scheduled downtime
2. Cost of implementing custom synchronization
3. Risks of data sync problems

Variations & Combinations



1. Cost/acceptability of scheduled downtime
2. Cost of implementing custom synchronization
3. Risks of data sync problems
4. Cost/acceptability short term performance issues

Variations & Combinations



Scheduling Your Migration

- In Introduction (10 minutes)
- Ev Initial Evaluation (25 minutes)
- Pl Migration Planning (20 minutes)
- Sc **Scheduling Your Migration** (15 minutes)
- Pm Post Migration / Post Mortem (10 minutes)
- QA Q&A (10 minutes)

Migration Schedule

- Before the Migration:
 - Instrument your entire system
 - Establish baselines
 - Create acceptance criteria from the baselines
 - Perform all planned system-level pre-migration refactorings
 - Reconfirm baselines post refactoring

Migration Schedule

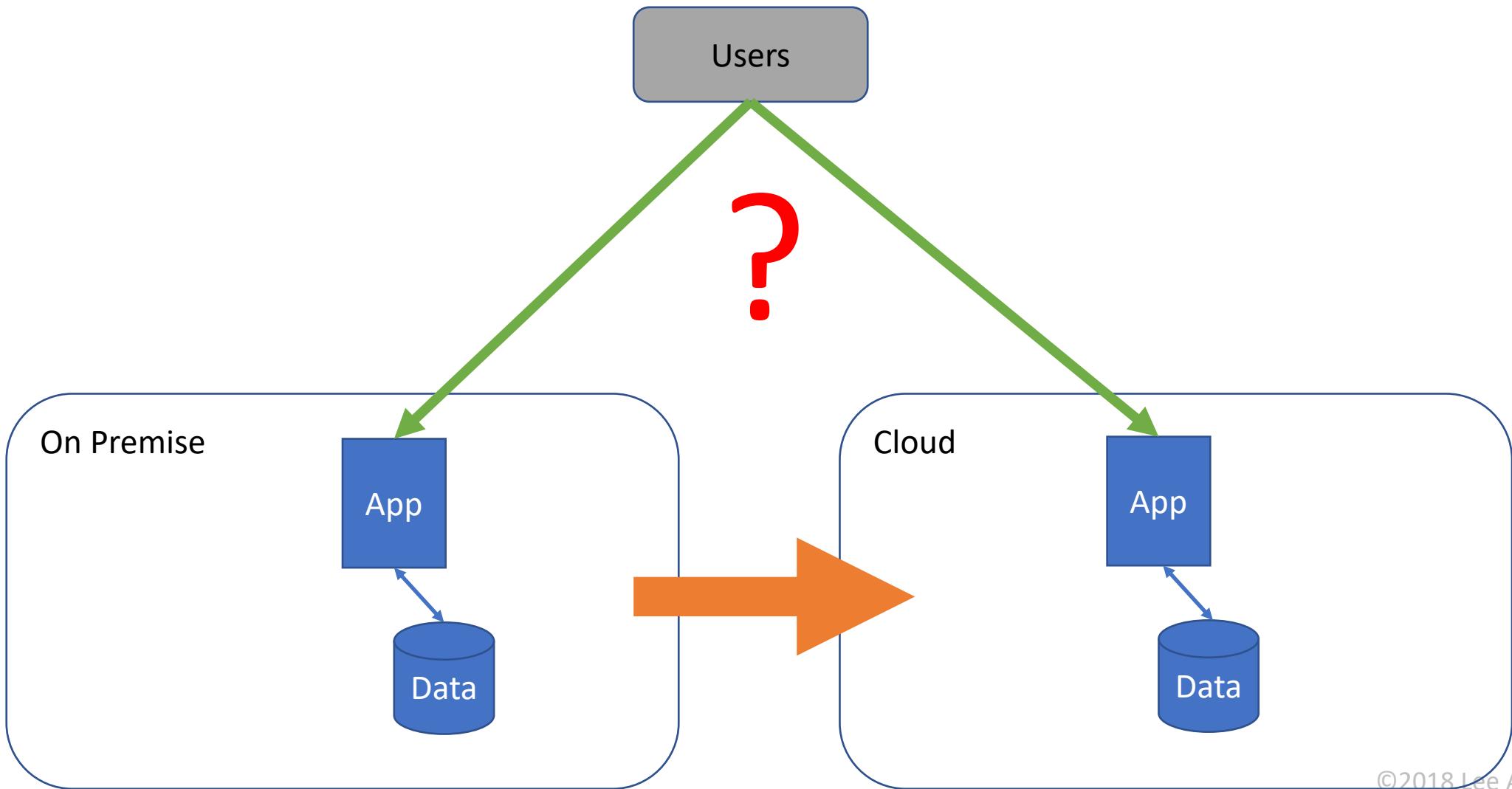
- Each Module/Service Being Migrated:
 - Do service-level pre-migration refactorings
 - Migrate data
 - Migrate the service
 - Resolve roadblocks/problems
 - Post-migration refactorings
 - Validate performance & acceptance criteria
 - Resolve any performance issues

Migration Schedule

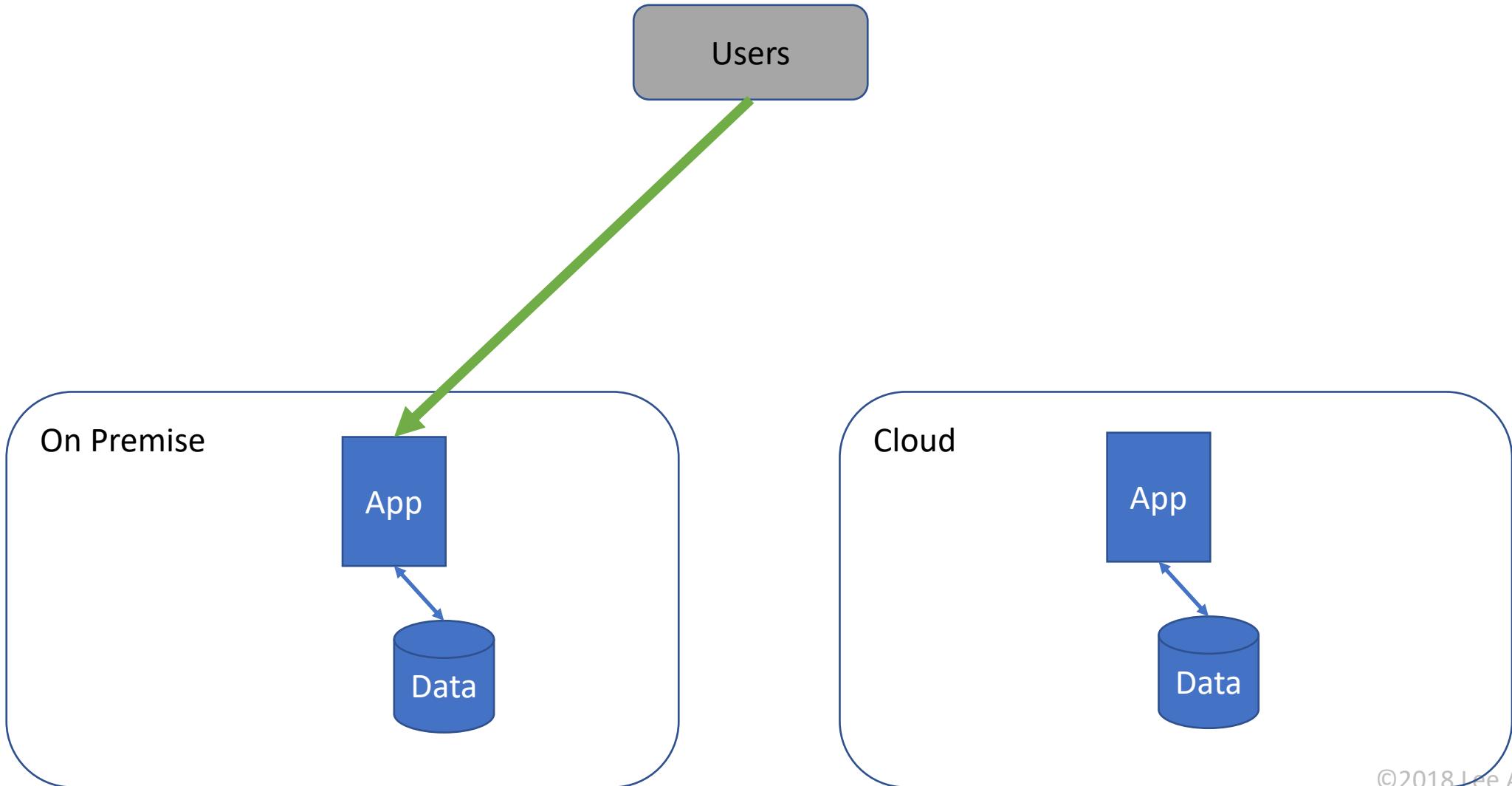
- After the Migration:
 - Planned post-migration system-level refactorings
 - Validate performance & acceptance criteria
 - Resolve any performance issues

Switching Over Production

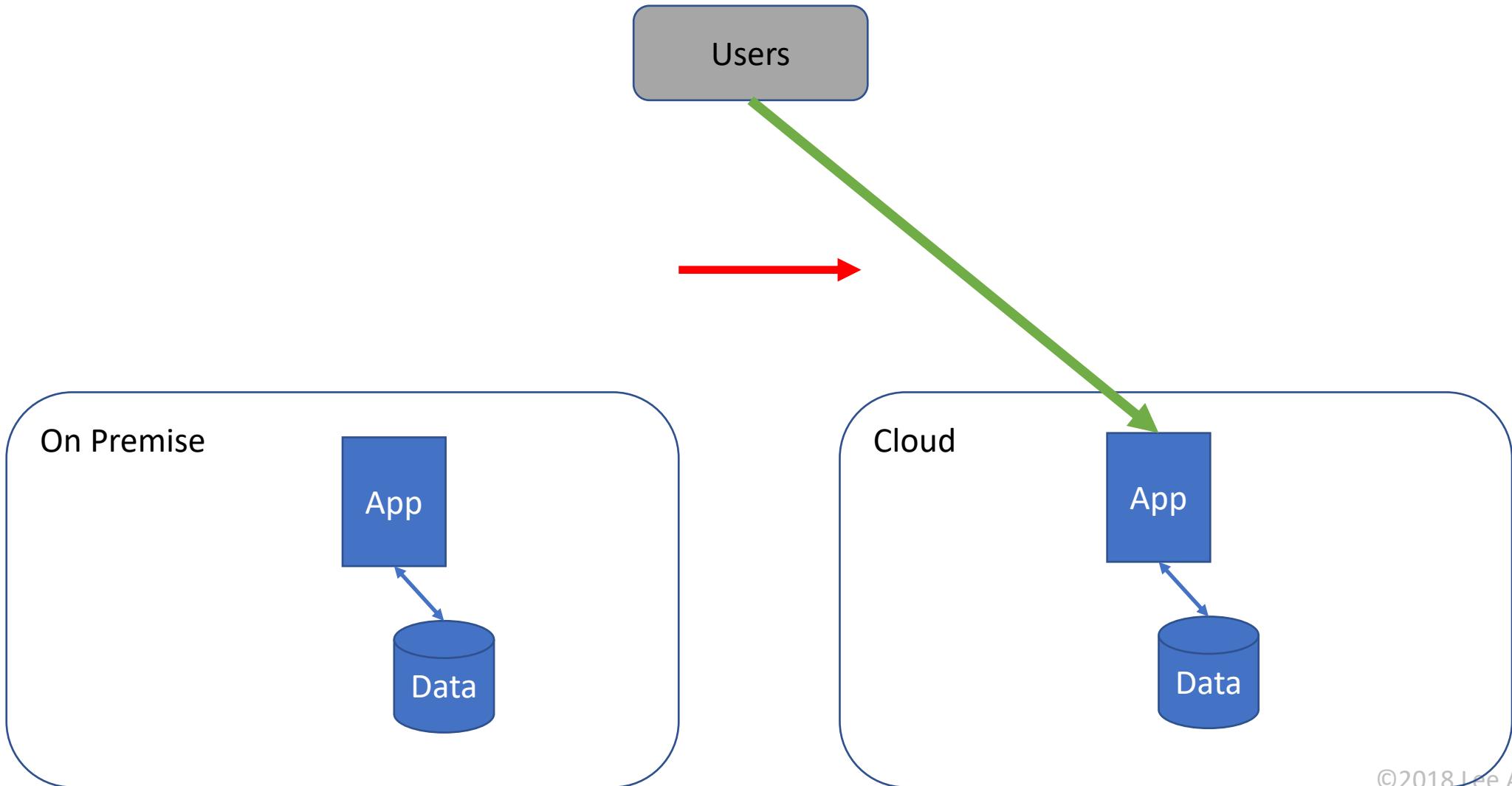
When Do You Switch Production Traffic?



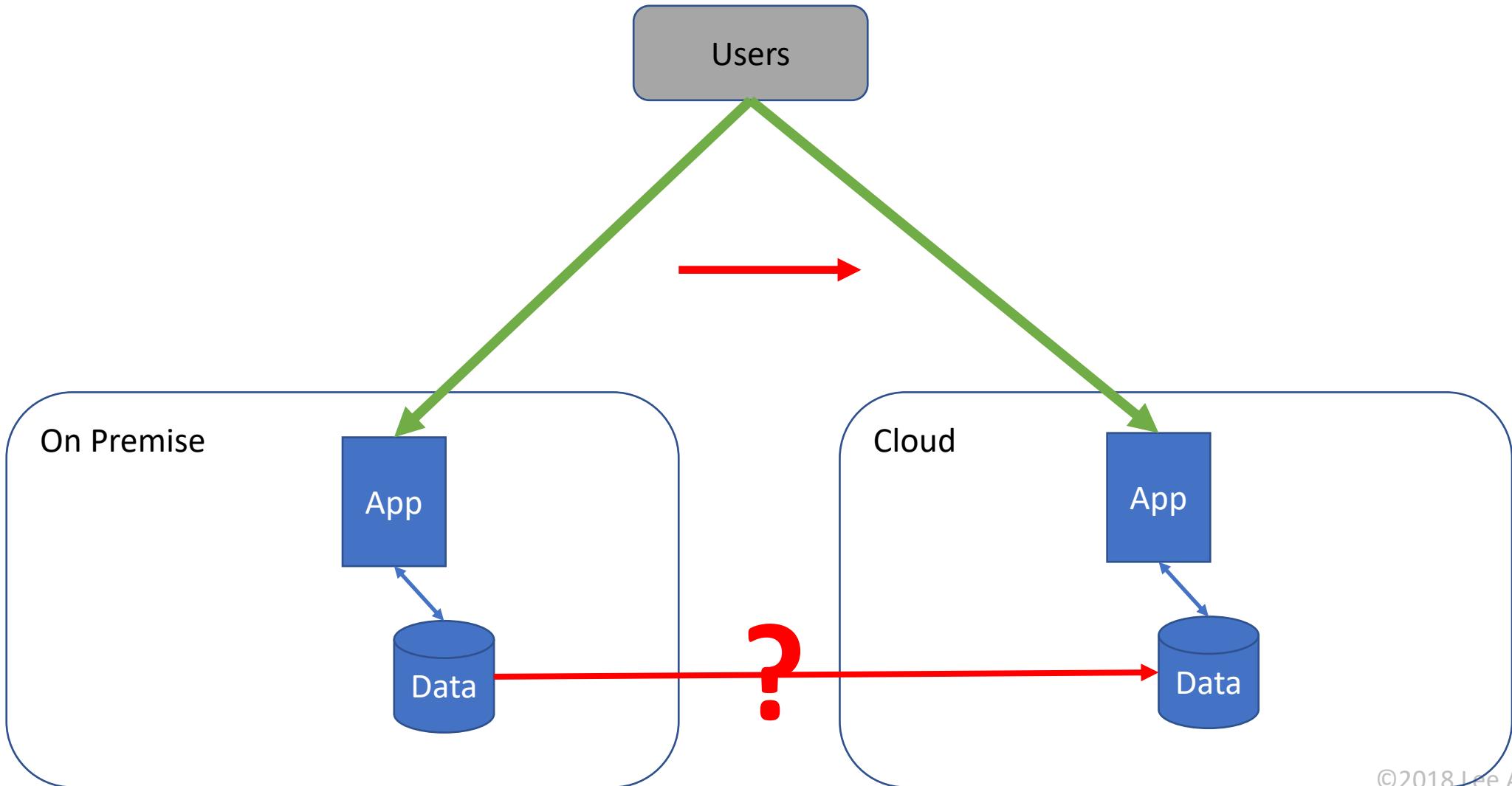
Switch Production: All At Once



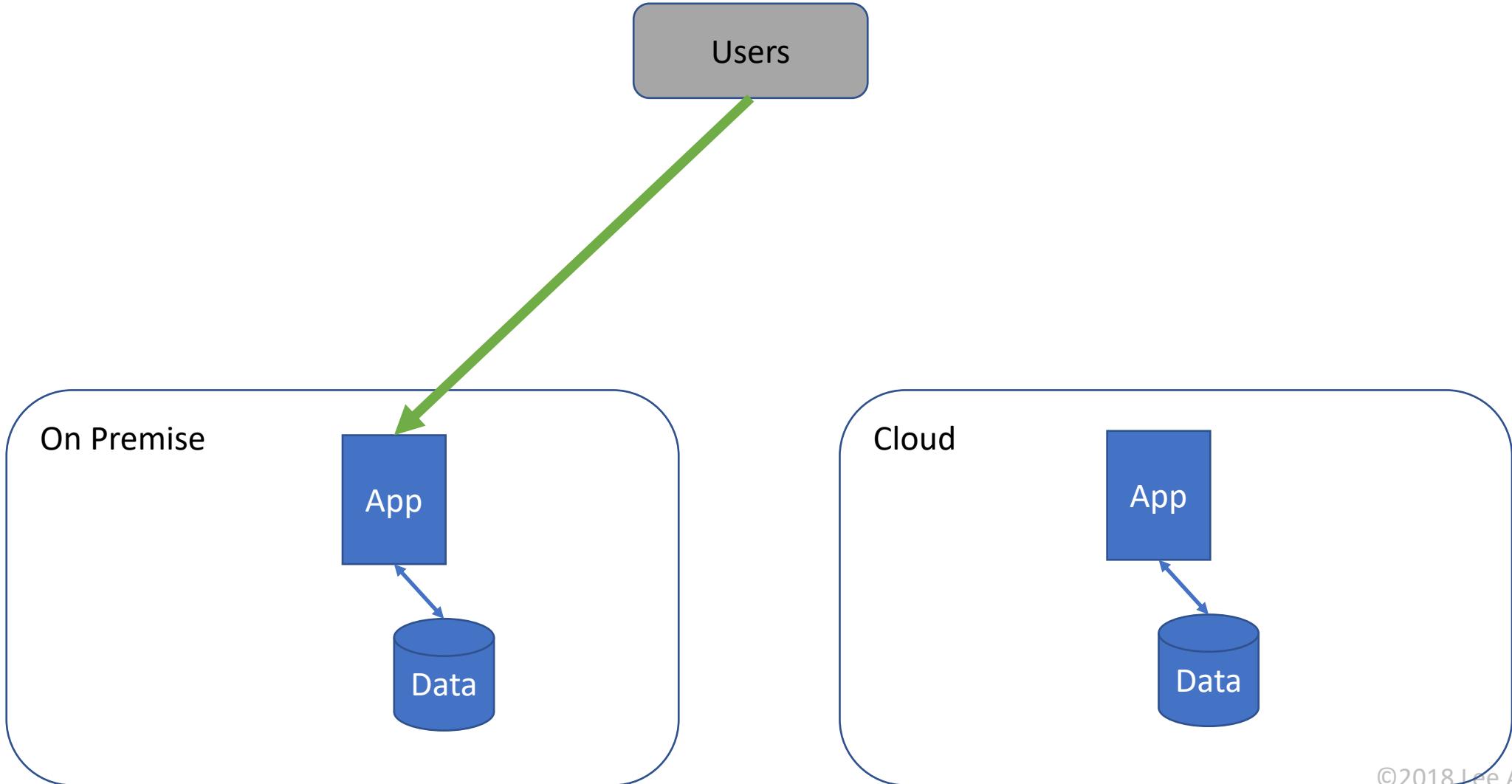
Switch Production: All At Once



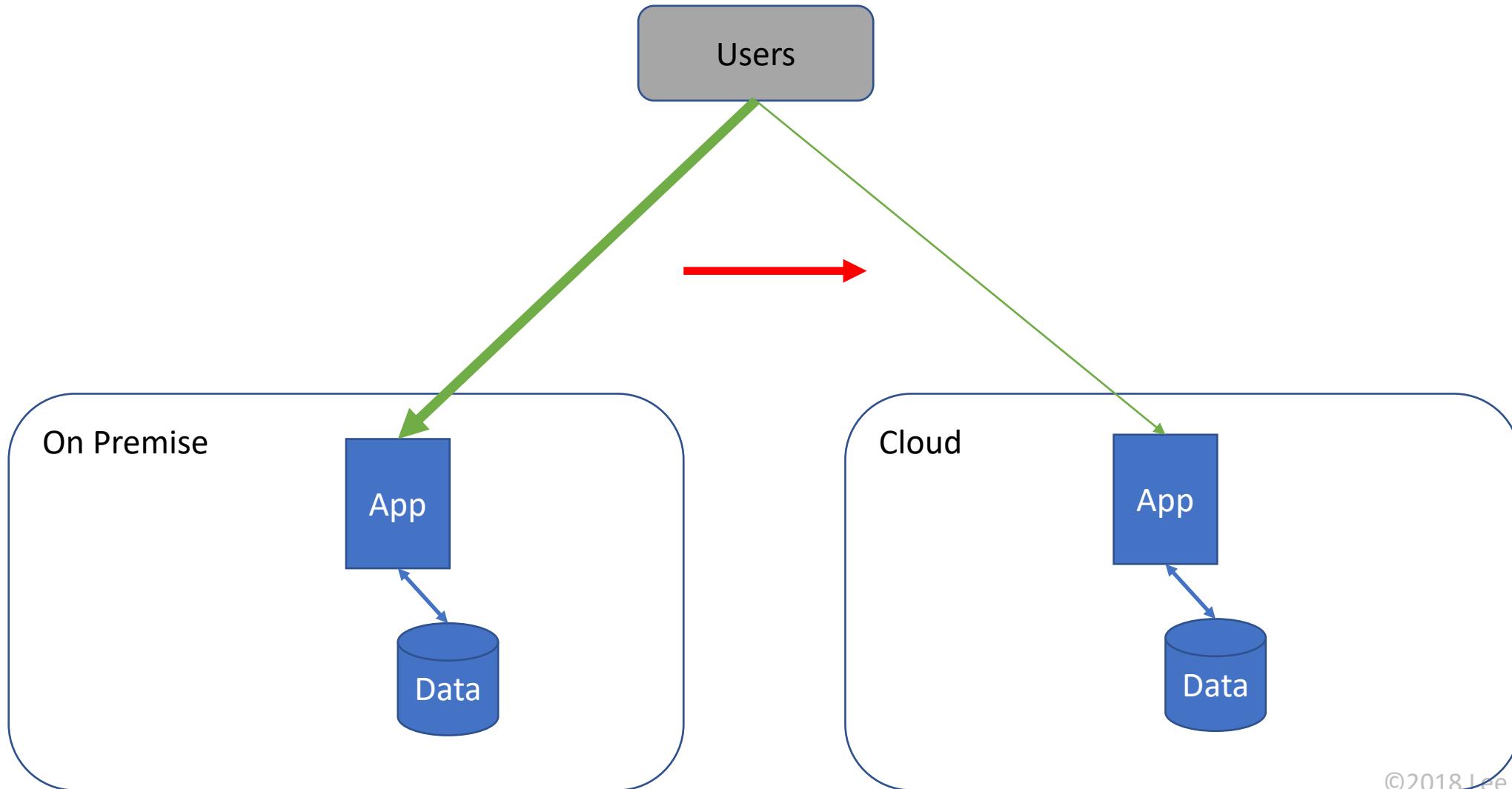
Switch Production: All At Once



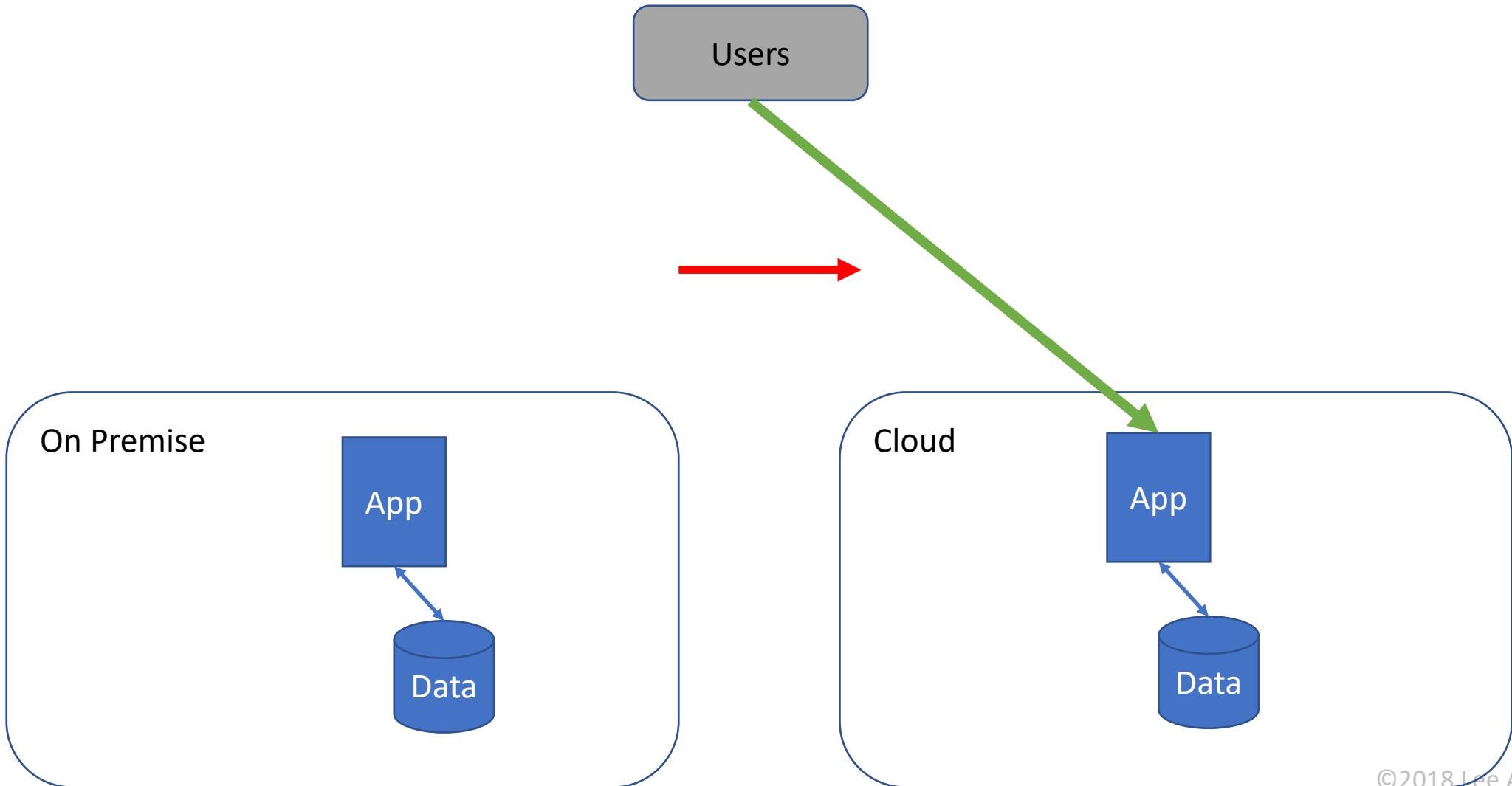
Switch Production: Ramped



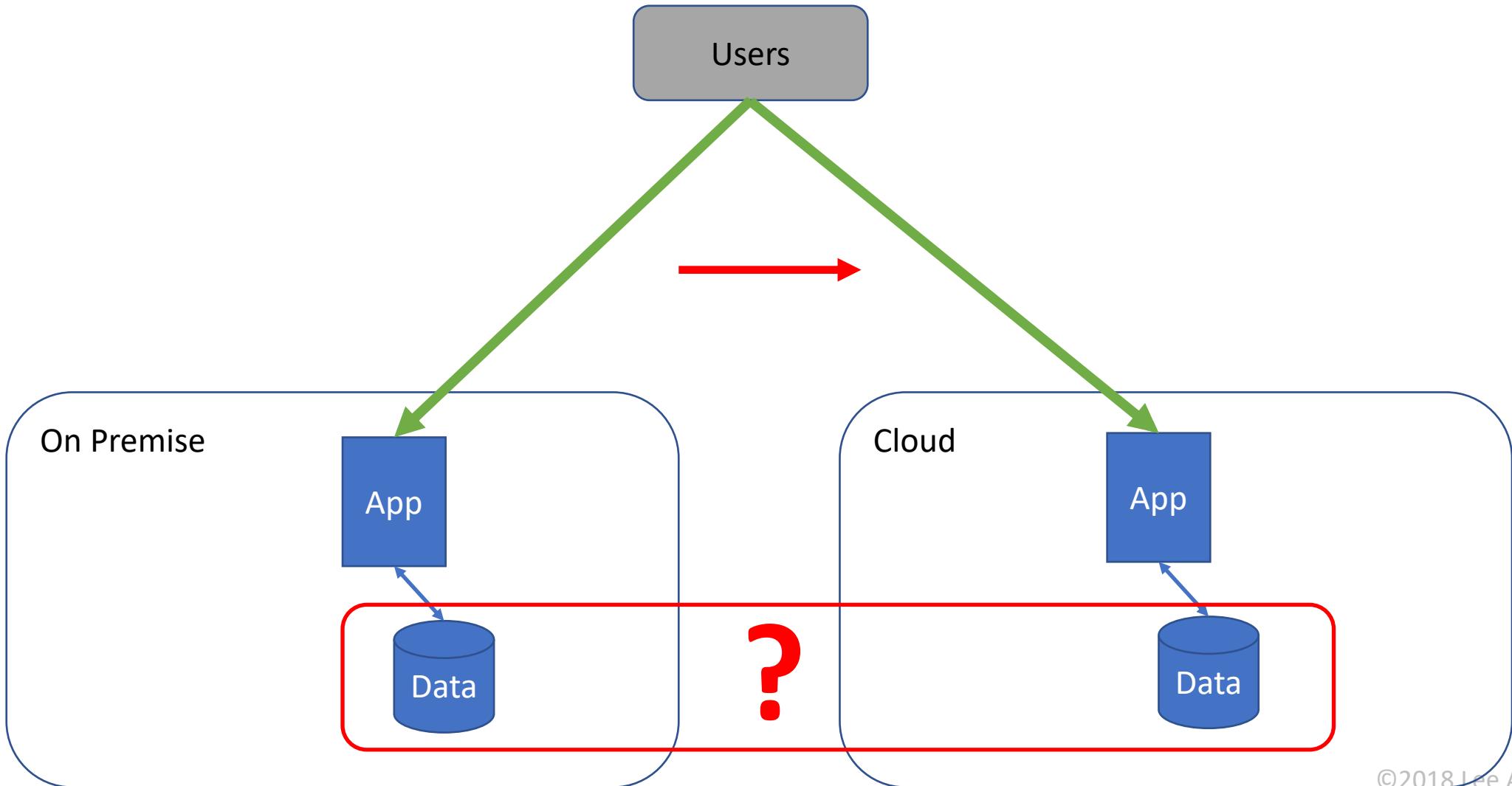
Switch Production: Ramped



Switch Production: Ramped



Switch Production: Ramped



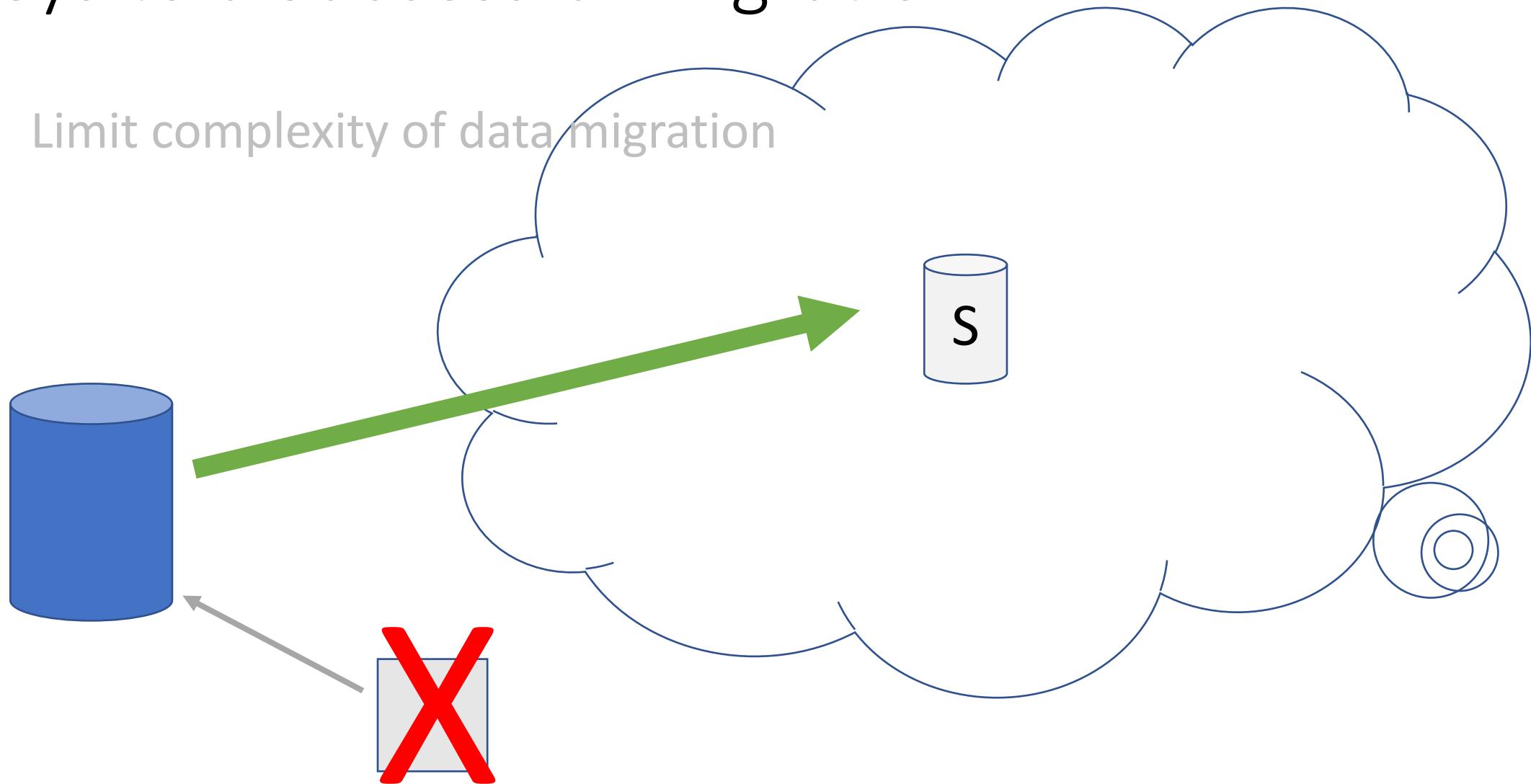
Keys to a Successful Migration

Keys to a Successful Migration

1. Limit complexity of data migration

Keys to a Successful Migration

1. Limit complexity of data migration

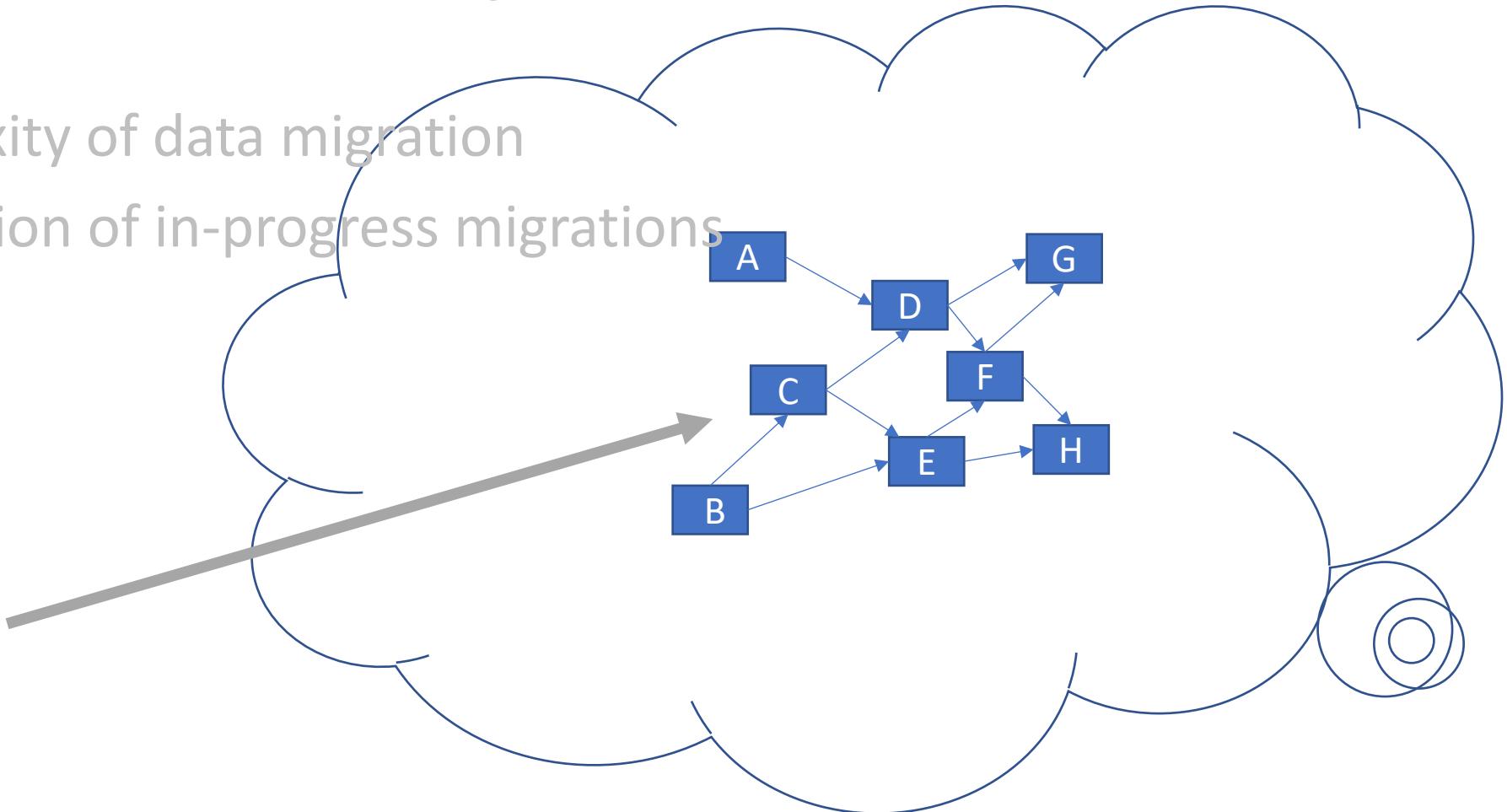
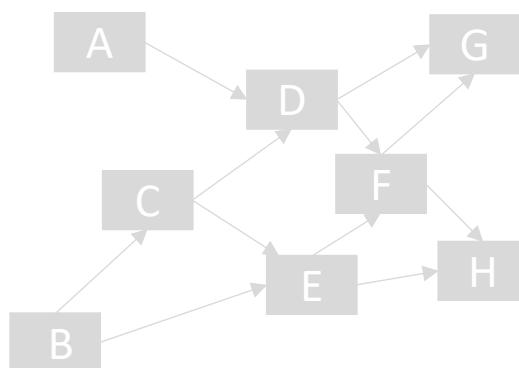


Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations

Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations

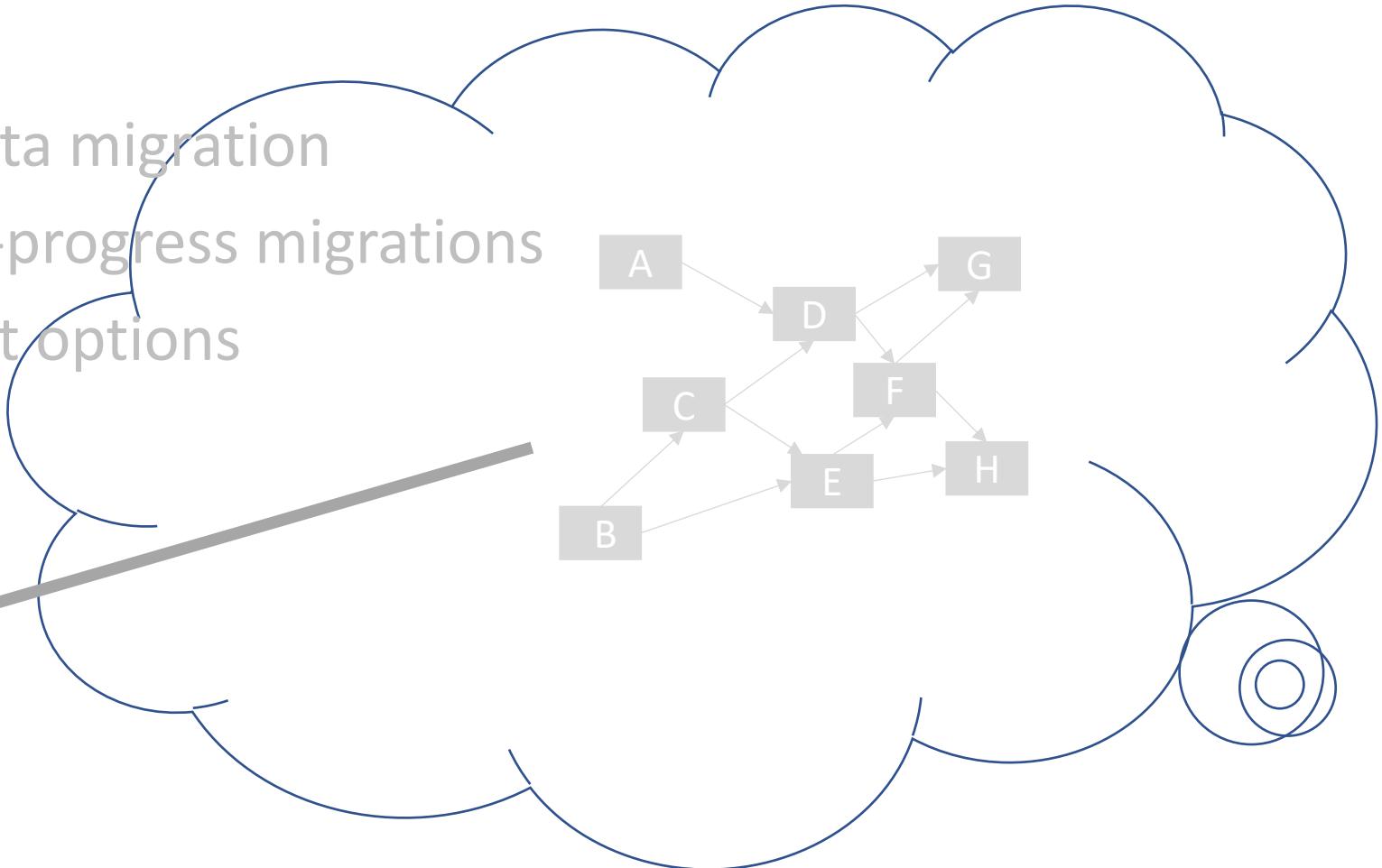
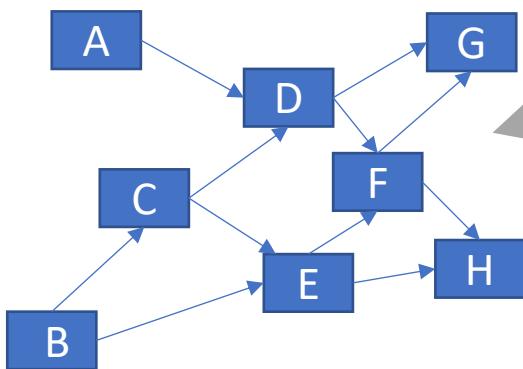


Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options

Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options

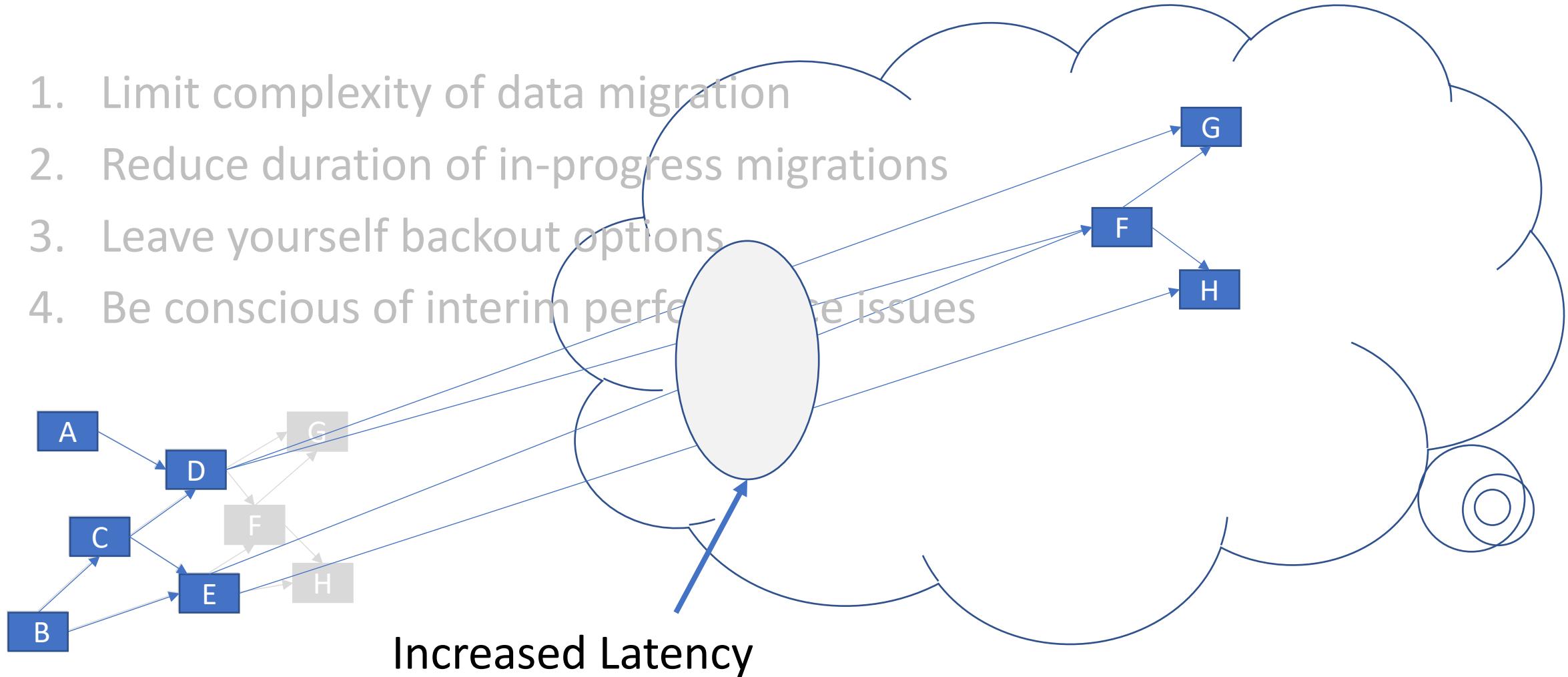


Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options
4. Be conscious of interim performance issues

Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options
4. Be conscious of interim performance issues



Keys to a Successful Migration

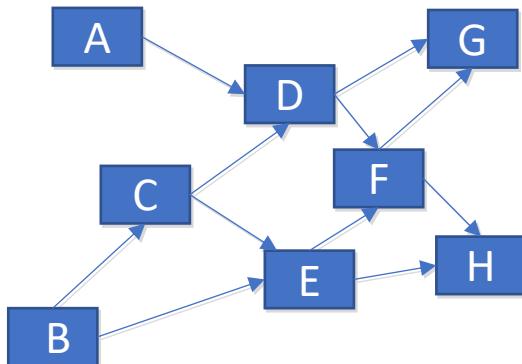
1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options
4. Be conscious of interim performance issues
5. Do refactoring before you migrate

Keys to a Successful Migration

- 
1. Limit complexity of data migration
 2. Reduce duration of in-progress migrations
 3. Leave yourself backout options
 4. Be conscious of interim performance issues
 5. Bring before you migrate

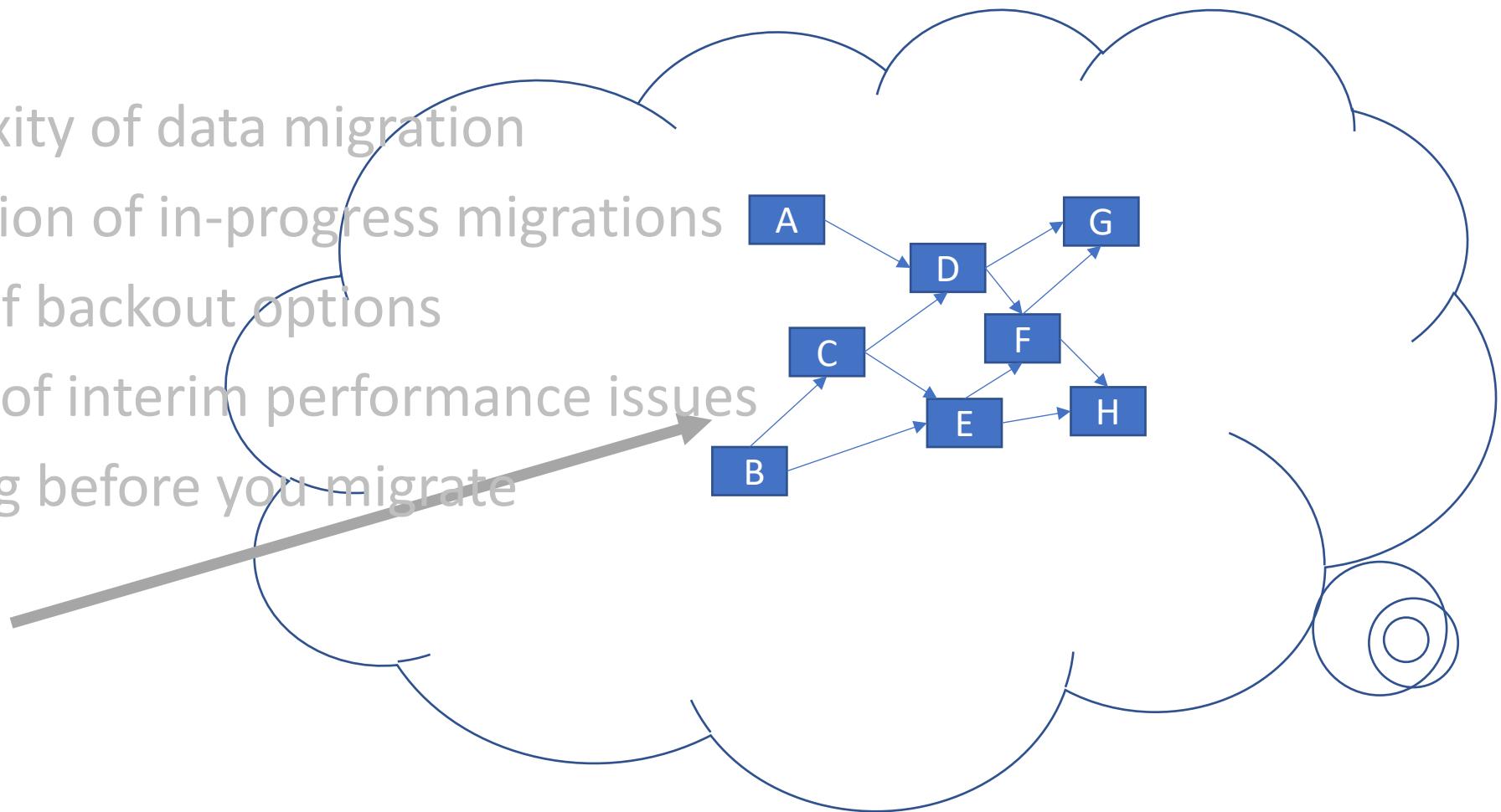
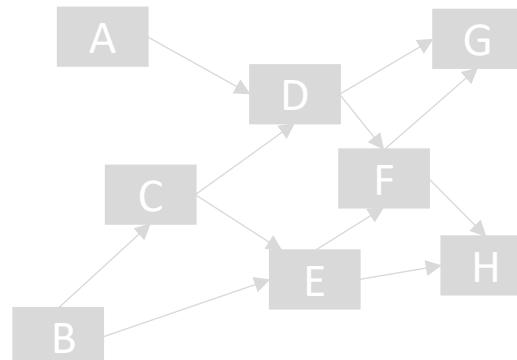
Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options
4. Be conscious of interim performance issues
5. Do refactoring before you migrate



Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options
4. Be conscious of interim performance issues
5. Do refactoring before you migrate



Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options
4. Be conscious of interim performance issues
5. Do refactoring before you migrate

Question: What challenges do you see in your specific migrations that I did not mention here?

Worksheet – Migration Steps

Worksheet - Migration Plan (Page 1)

- Application Info
 - Migration Architect
- Style of Migration
- Cloud Ready Analysis
- Data Migration

<p>Worksheet: Migration Plan</p> <p>Application: _____</p> <p>Migration Architect: _____</p> <p>Style of Migration</p> <p><input type="checkbox"/> All at once <input type="checkbox"/> One component at a time <input type="checkbox"/> Inside Out <input type="checkbox"/> Outside In</p> <p>Notes:</p>	<p>Cloud Ready Analysis</p> <p><i>What refactoring needs to happen before a migration is initiated?</i></p> <p>Cloud Ready Analysis</p> <ul style="list-style-type: none">• Will it run in the cloud (virtual servers)?• Maintain state within the app itself?<ul style="list-style-type: none">◦ aka, filesystem/disk• Custom network settings?• Unique load balancing?• Twelve Factor App https://12factor.net• Application Performance Monitoring (APM)	<p>Data Migration</p> <p><i>What data needs to be migrated? What strategy will you use? What about offline data transfer such as AWS Snowball?</i></p> <p>Data Migration</p> <ul style="list-style-type: none">• Synchronized Masters• Master/Slave Switch• Read/Only Datasets• Large Dataset / Offline Transfer
---	--	---

Worksheet - Migration Plan (Page 2)

- Post Cloud Refactorings (Required)
- Post Migration Possibilities (Optional)
- Production Go-Live Plans

Post Cloud Refactorings (Required)

What refactoring must happen after application has moved to the cloud, but are important enough that you will implement them as part of the migration project? This includes required refactorings for including cloud-based services into the project, performance tunings, and usage of dynamic resource elements.

- | Required Cloud Changes |
|---|
| <ul style="list-style-type: none">• Required performance tuning• Cost containment• Database changes<ul style="list-style-type: none">◦ Start using S3, DynamoDB...• Resource management• CI/CD changes• Operational procedures |

Post Migration Possibilities (Optional)

What refactoring would be useful after the migration is complete, but not strictly required? This includes useful but not required refactorings to include cloud-based services into the project, performance tunings, and usage of dynamic resources allocations.

- | Optional Cloud Changes |
|---|
| <ul style="list-style-type: none">• Optional performance tuning• Cost optimizations• Dynamic services<ul style="list-style-type: none">◦ Such as Lambda...• Cloud datastore usage/patterns<ul style="list-style-type: none">◦ Increased use of S3, DynamoDB... |

Production Go-Live Plans

What are your plans for switching production traffic from on-prem stack to cloud stack?

[] All at once [] Slow Ramp [] Canary Test

Load Shift Strategy

Notes

- | Production Go-Live |
|--|
| <ul style="list-style-type: none">• All at once
(0%→100% switch)• Slow ramp
(0%→100% over period of time)• Canary Test
(0%-n%, to test on small sample)• Strategy for load shifting
(load balancer, DNS, ...) |

Worksheet - Migration Plan (Page 3)

- Reference Material:
 - Migration Steps
 - Before
 - During
 - After
 - Keys to a Successful Migration

Reference		
Migration Steps		
Before Migration	During Migration	After Migration
<ul style="list-style-type: none">• Instrument your entire system• Establish baselines• Create acceptance criteria from the baselines• Perform all planned system-level pre-migration refactorings• Reconfirm baselines post refactoring	<ul style="list-style-type: none">• Do service-level pre-migration refactorings• Migrate data• Migrate the service• Resolve roadblocks/problems• Post-migration refactorings• Validate performance & acceptance criteria• Resolve any performance issues	<ul style="list-style-type: none">• Planned post-migration system-level refactorings• Validate performance & acceptance criteria• Resolve any performance issues
Keys to a Successful Migration		
<ol style="list-style-type: none">1. Limit complexity of data migration2. Reduce duration of in-progress migrations3. Leave yourself backout options4. Be conscious of interim performance issues5. Do refactoring before you migrate		

Worksheet - Migration Plan (Page 4)

- Migration Steps
planning page

Migration Steps

Use this space to plan out the steps of your migration, using the information above as a reference.

Post Migration / Post Mortem

- In **Introduction** (10 minutes)
- Ev **Initial Evaluation** (25 minutes)
- Pl **Migration Planning** (20 minutes)
- Sc **Scheduling Your Migration** (15 minutes)
- Pm **Post Migration / Post Mortem** (10 minutes)
- QA **Q&A** (10 minutes)

Resource Optimization

Resource Pool

Resource Pool

- Dynamic, on-demand resources
 - Production scaling
 - Development, Testing

Resource Pool

- Dynamic, on-demand resources
 - Production scaling
 - Development, Testing
- Geographic Scaling
 - Region specific datacenters
 - Backup/redundant datacenters

Resource Pool

- Dynamic, on-demand resources
 - Production scaling
 - Development, Testing
- Geographic Scaling
 - Region specific datacenters
 - Backup/redundant datacenters
- Easy experimentation

Resource Pool

- Dynamic, on-demand resources
 - Production scaling
 - Development, Testing
- Geographic Scaling
 - Region specific datacenters
 - Backup/redundant datacenters
- Easy experimentation

Key:

Well architected cloud application

Cloud Costing

Cloud Costing

Cloud Costing

“The cloud is *cheaper* than on-prem...”

Cloud Costing

“The cloud is *cheaper* than on-prem...”

“The cloud is *more expensive* than on-prem...”

Inflated Cloud Infrastructure Costs

- Using cloud in a static manner

Inflated Cloud Infrastructure Costs

- Using cloud in a static manner
- Hidden on-prem costs exposed in cloud

Inflated Cloud Infrastructure Costs

- Using cloud in a static manner
- Hidden on-prem costs exposed in cloud
- Capital vs Expense

Inflated Cloud Infrastructure Costs

- Using cloud in a static manner
- Hidden on-prem costs exposed in cloud
- Capital vs Expense
- Utility pricing and reserved capacity

Cloud Cost Management

Cloud Cost Management

Architecture

Cloud Cost Management

Architecture

Dynamic

Resource

Utilization

Cloud Cost Management

Architecture

Dynamic

Resource

Utilization

- *Autoscaling*
- *Serverless, Dynamic Containers*

Cloud Cost Management

Architecture

Dynamic

Resource

Utilization

Operational

- *Autoscaling*
- *Serverless, Dynamic Containers*

Cloud Cost Management

Architecture

Dynamic

Resource

Utilization

Operational

Capacity

Planning

- *Autoscaling*
- *Serverless, Dynamic Containers*

Cloud Cost Management

Architecture

*Dynamic
Resource
Utilization*

- *Autoscaling*
- *Serverless, Dynamic Containers*

Operational

*Capacity
Planning*

- *Reserved Instances*
- *Understand billing*

Q&A

- In** Introduction (10 minutes)
- Ev** Initial Evaluation (25 minutes)
- Pl** Migration Planning (20 minutes)
- Sc** Scheduling Your Migration (15 minutes)
- Pm** Post Migration / Post Mortem (10 minutes)
- QA** Q&A (10 minutes)

Q&A



Thank You

Lee Atchison, Senior Director Strategic Architecture, New Relic

www.leeatchison.com

Twitter: @leeatchison

LinkedIn: leeatchison