Data Patterns for Successful Applications

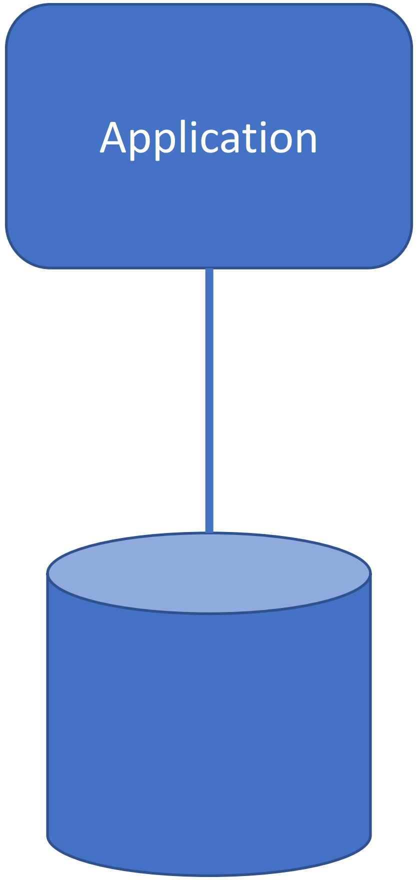
Sean Whitesell

So, what we going to do?

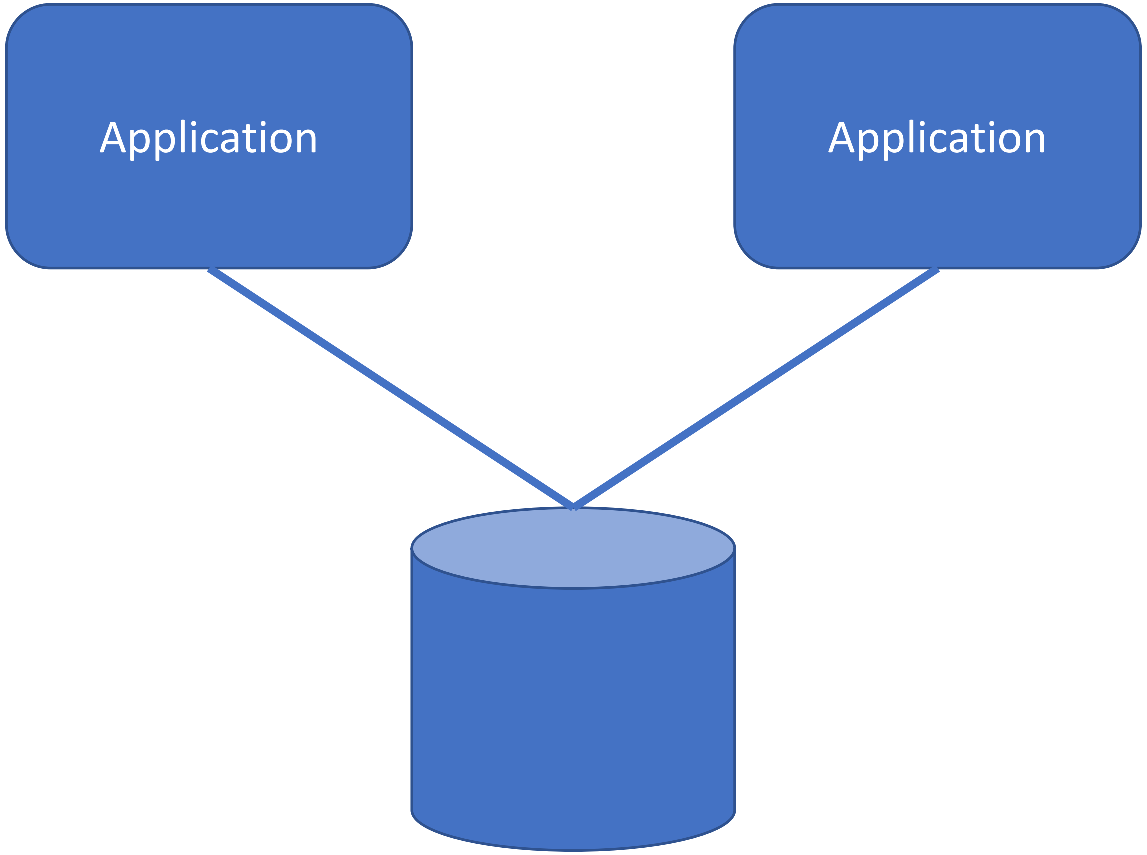
Agenda:

- Common Architecture
- Data Model
- CQS
- CQRS
- Eventual Consistency
- Views
- Event Sourcing
- CQRS + Event Sourcing

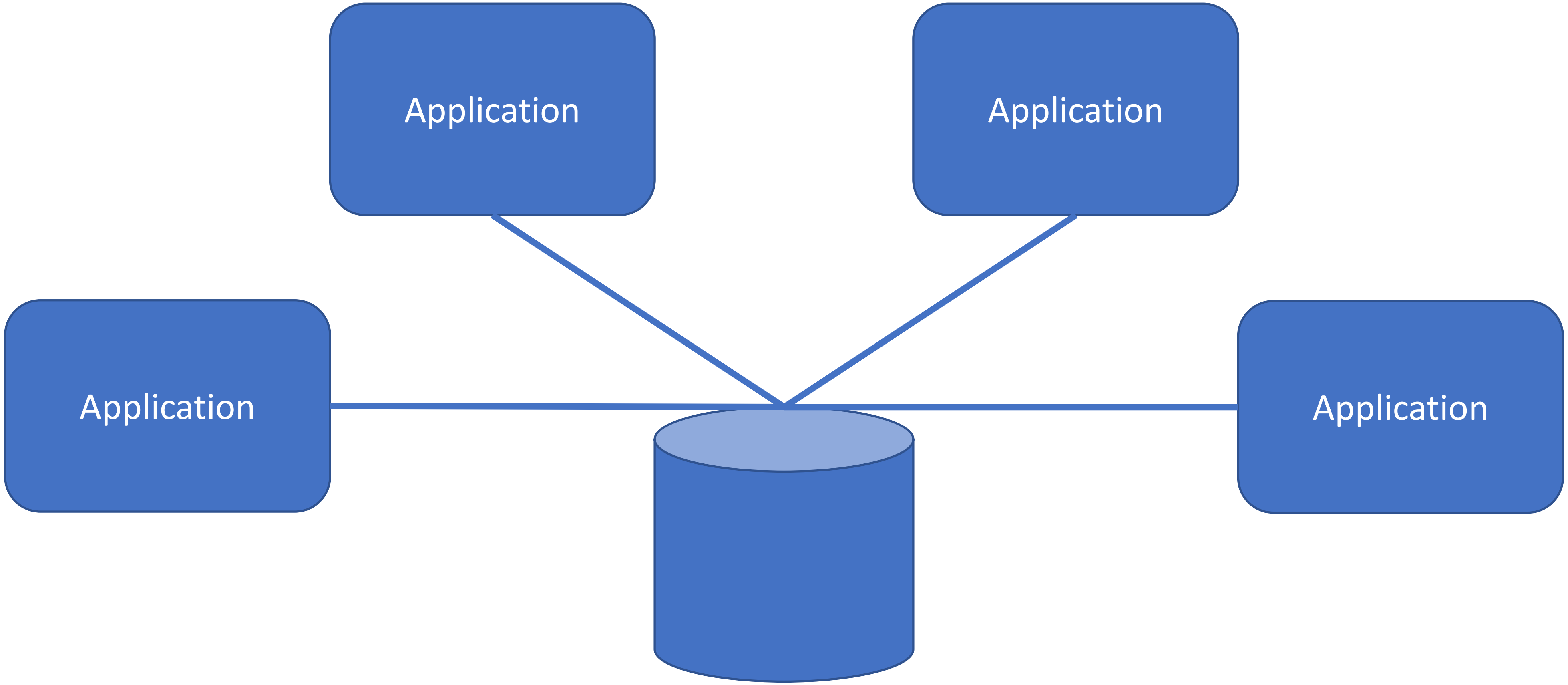
Common Architecture



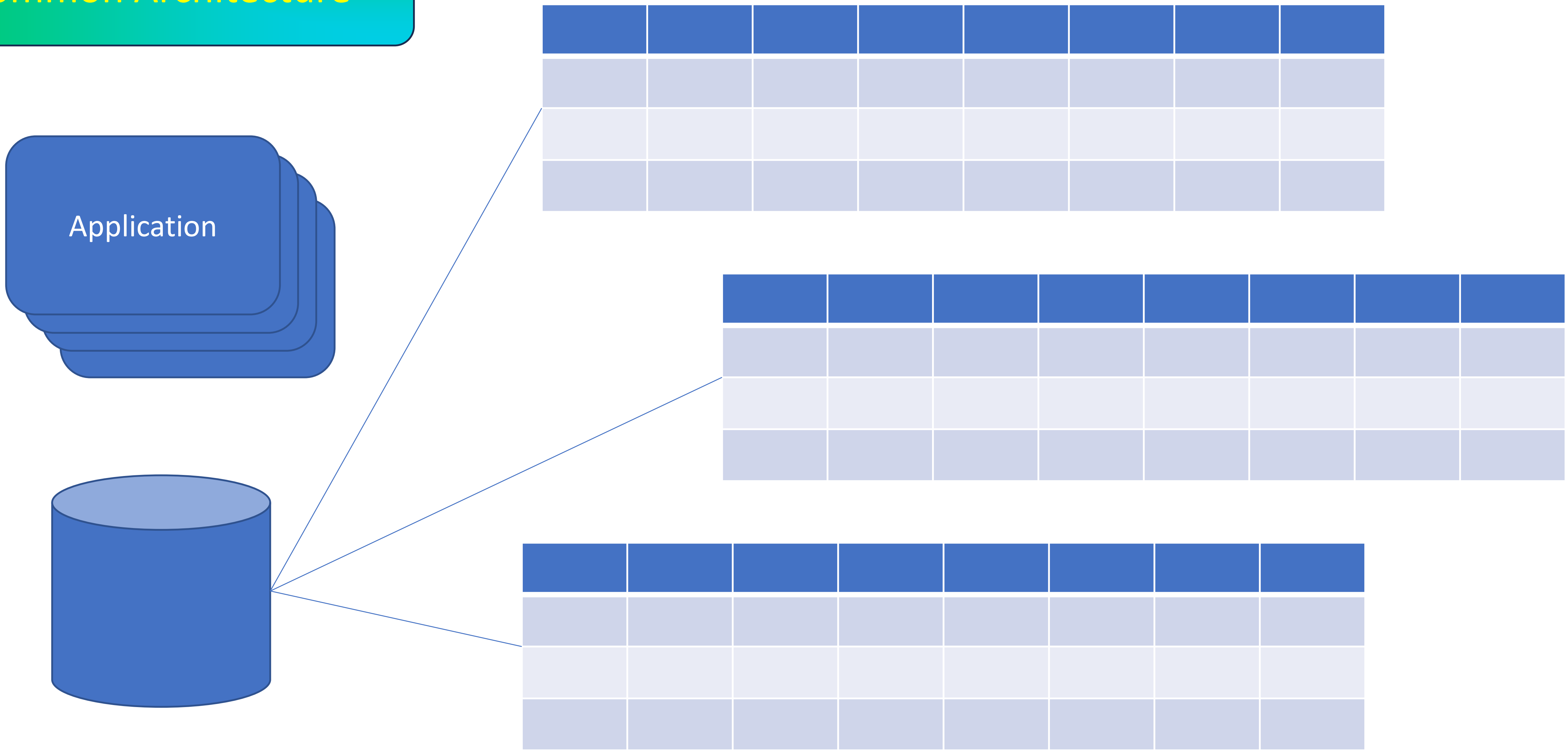
Common Architecture



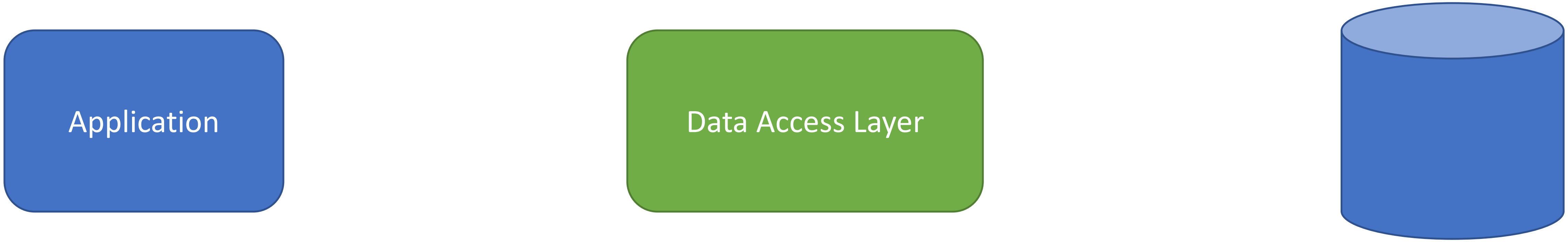
Common Architecture



Common Architecture



Common Architecture

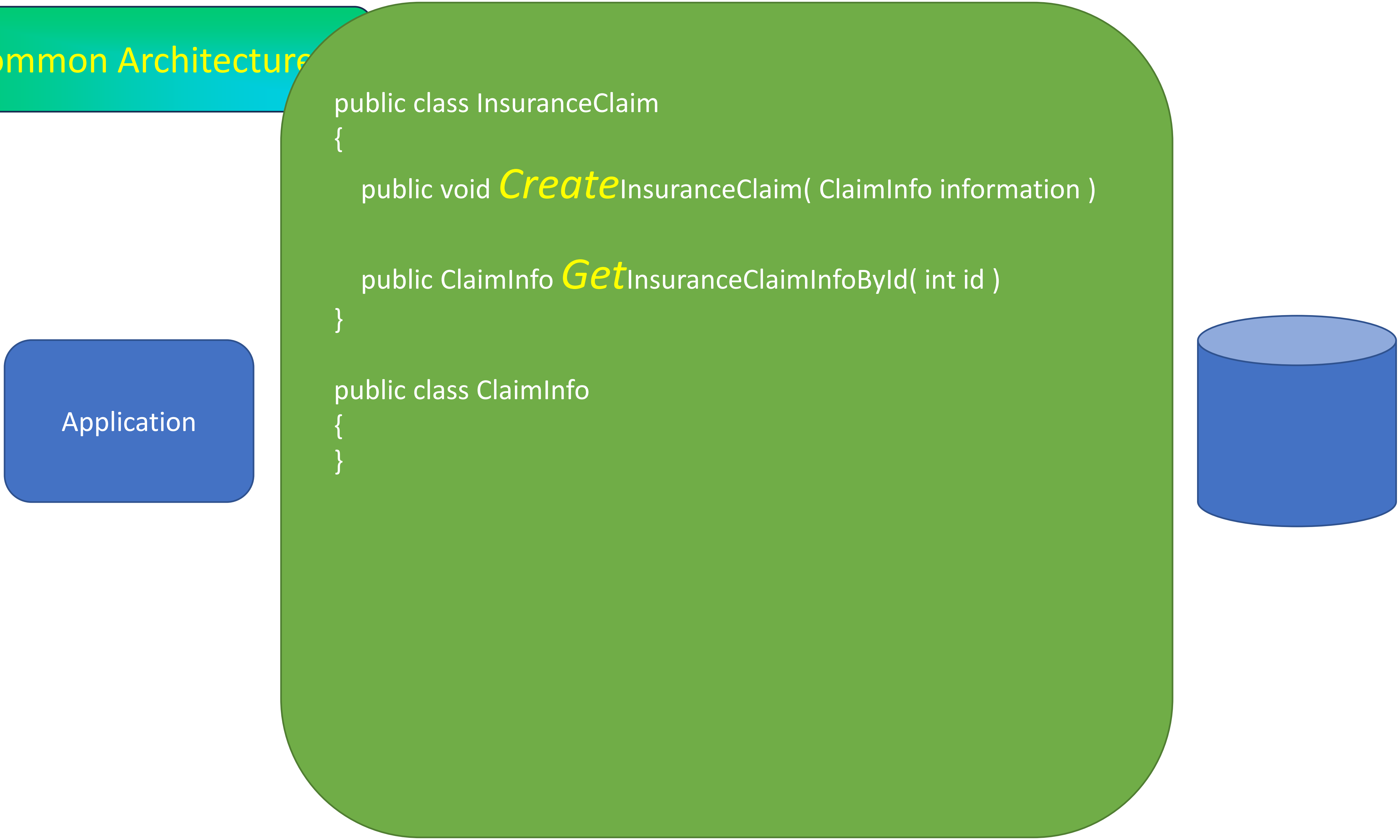


Application

Data Access Layer

Could be inside the app or in a DLL/package.

Common Architecture

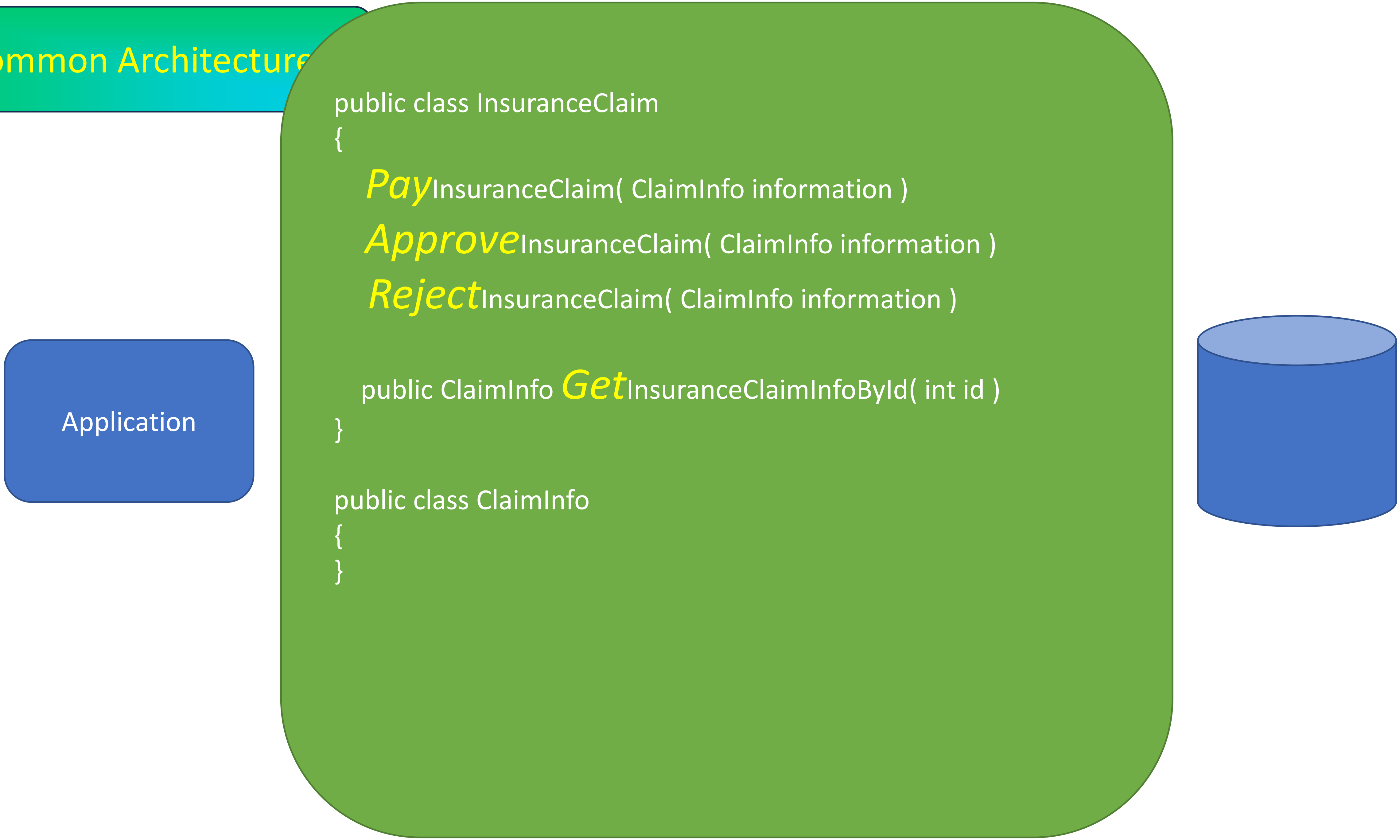


```
public class InsuranceClaim
{
    public void CreateInsuranceClaim( ClaimInfo information )

    public ClaimInfo GetInsuranceClaimInfoById( int id )
}

public class ClaimInfo
{
}
```

Common Architecture

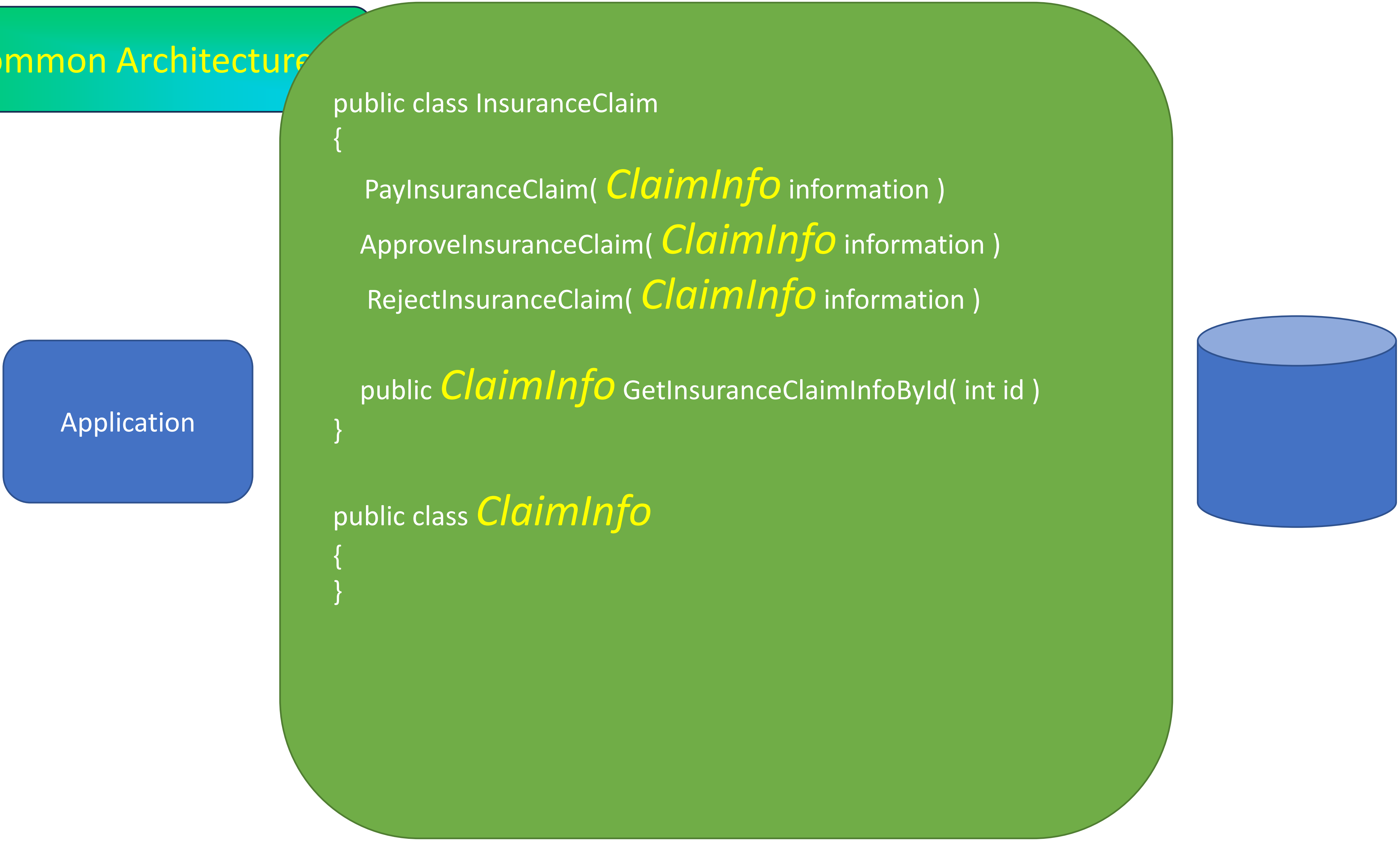


```
public class InsuranceClaim
{
    PayInsuranceClaim( ClaimInfo information )
    ApproveInsuranceClaim( ClaimInfo information )
    RejectInsuranceClaim( ClaimInfo information )

   public ClaimInfo GetInsuranceClaimInfoById( int id )
}

public class ClaimInfo
{
}
```

Common Architecture

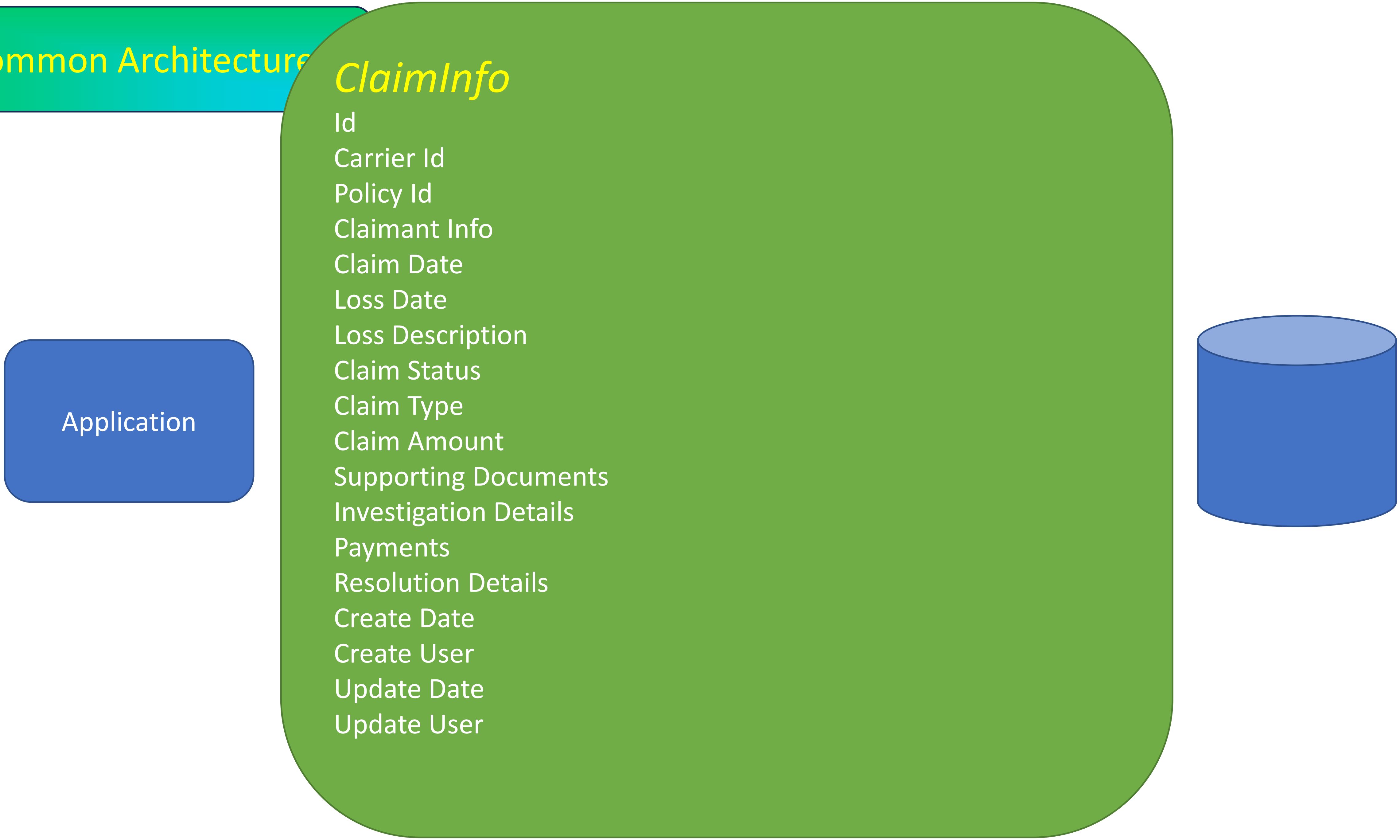


```
public class InsuranceClaim
{
    PayInsuranceClaim( ClaimInfo information )
    ApproveInsuranceClaim( ClaimInfo information )
    RejectInsuranceClaim( ClaimInfo information )

    public ClaimInfo GetInsuranceClaimInfoById( int id )
}

public class ClaimInfo
{
}
```

Common Architecture

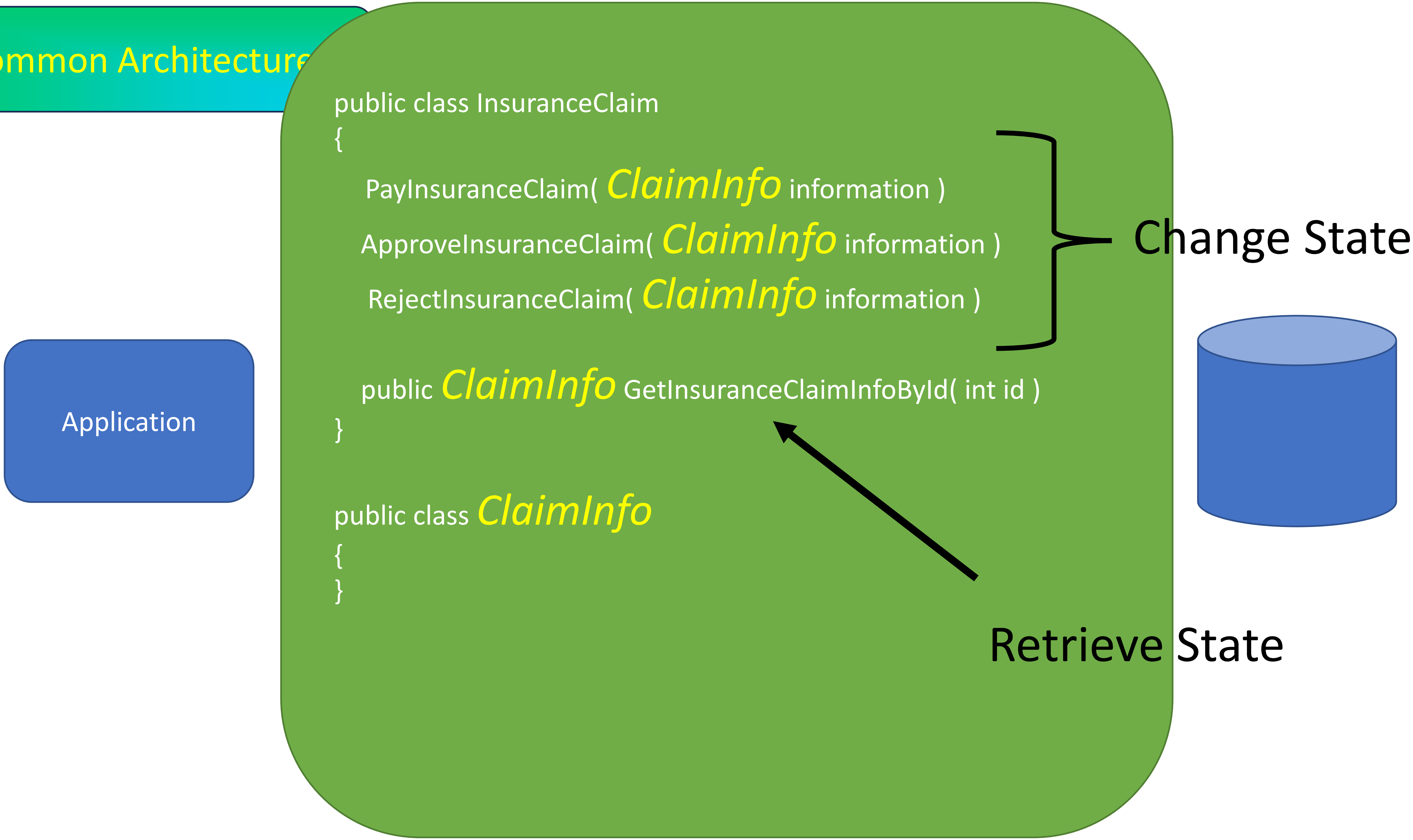


Application

ClaimInfo

- Id
- Carrier Id
- Policy Id
- Claimant Info
- Claim Date
- Loss Date
- Loss Description
- Claim Status
- Claim Type
- Claim Amount
- Supporting Documents
- Investigation Details
- Payments
- Resolution Details
- Create Date
- Create User
- Update Date
- Update User

Common Architecture



```
public class InsuranceClaim
{
    PayInsuranceClaim( ClaimInfo information )
    ApproveInsuranceClaim( ClaimInfo information )
    RejectInsuranceClaim( ClaimInfo information )

    public ClaimInfo GetInsuranceClaimInfoById( int id )
}

public class ClaimInfo
{
}
```

CQS

Command – Query – Separation

Common Architecture

```
public class InsuranceClaim
{
    PayInsuranceClaim( ClaimInfo information )
    ApproveInsuranceClaim( ClaimInfo information )
    RejectInsuranceClaim( ClaimInfo information )

}
```

Change State

Retrieve State

Common Architecture

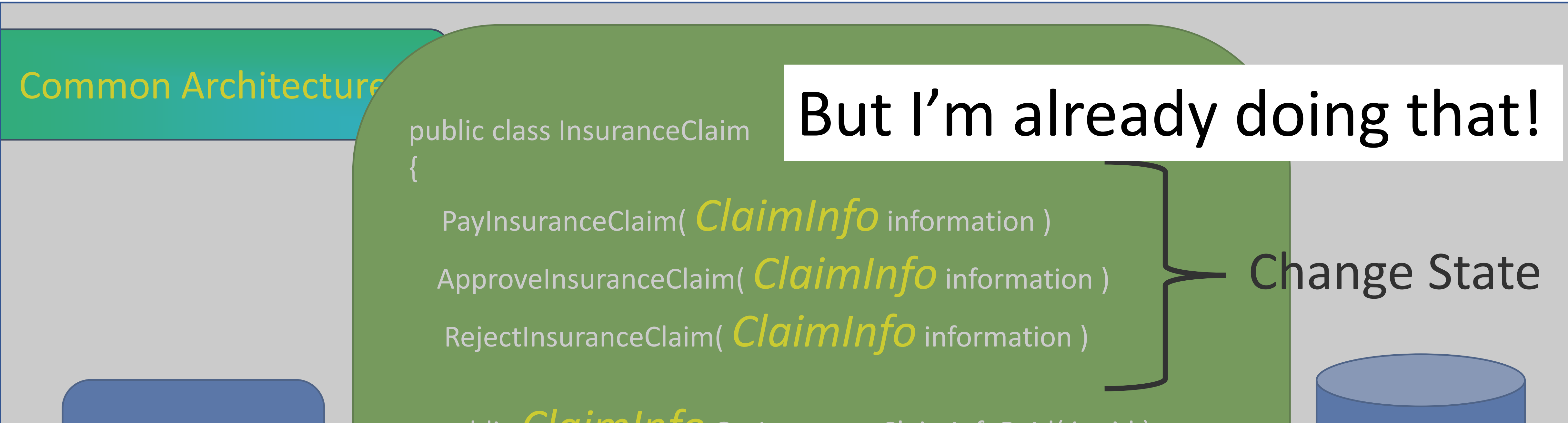
```
public class InsuranceClaim
{
    PayInsuranceClaim( ClaimInfo information )
    ApproveInsuranceClaim( ClaimInfo information )
    RejectInsuranceClaim( ClaimInfo information )
```

Change State

CQS

Command – Query – Separation

Retrieving data should not cause data to be changed.



```
public class InsuranceClaim
{
    PayInsuranceClaim( ClaimInfo information )
    ApproveInsuranceClaim( ClaimInfo information )
    RejectInsuranceClaim( ClaimInfo information )
```

But I’m already doing that!

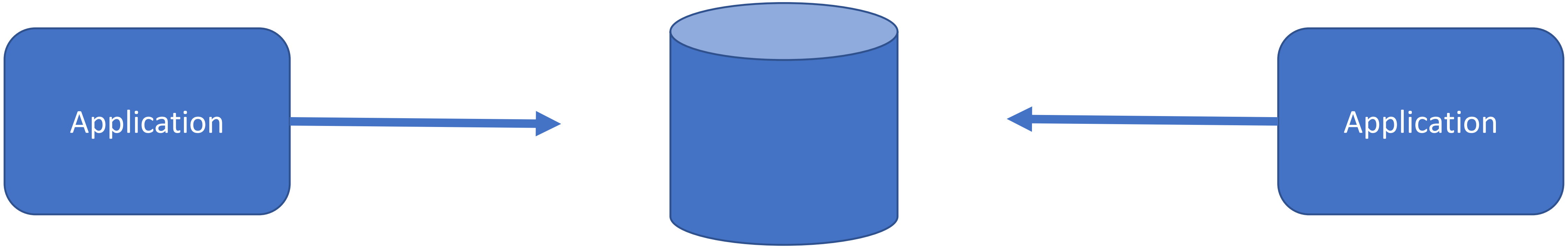
Change State

CQS

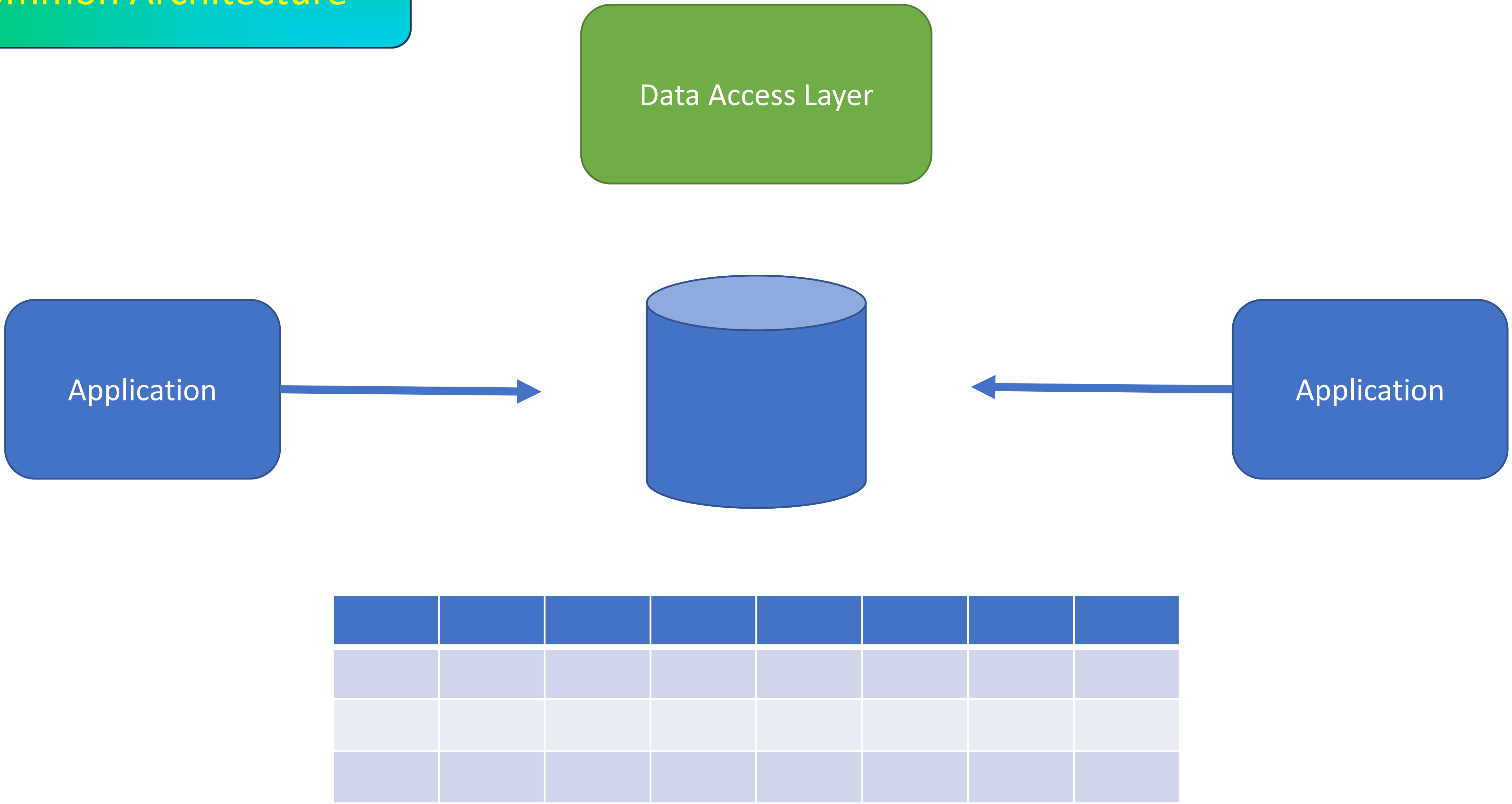
Command – Query – Separation

Retrieving data should not cause data to be changed.

Common Architecture



Common Architecture

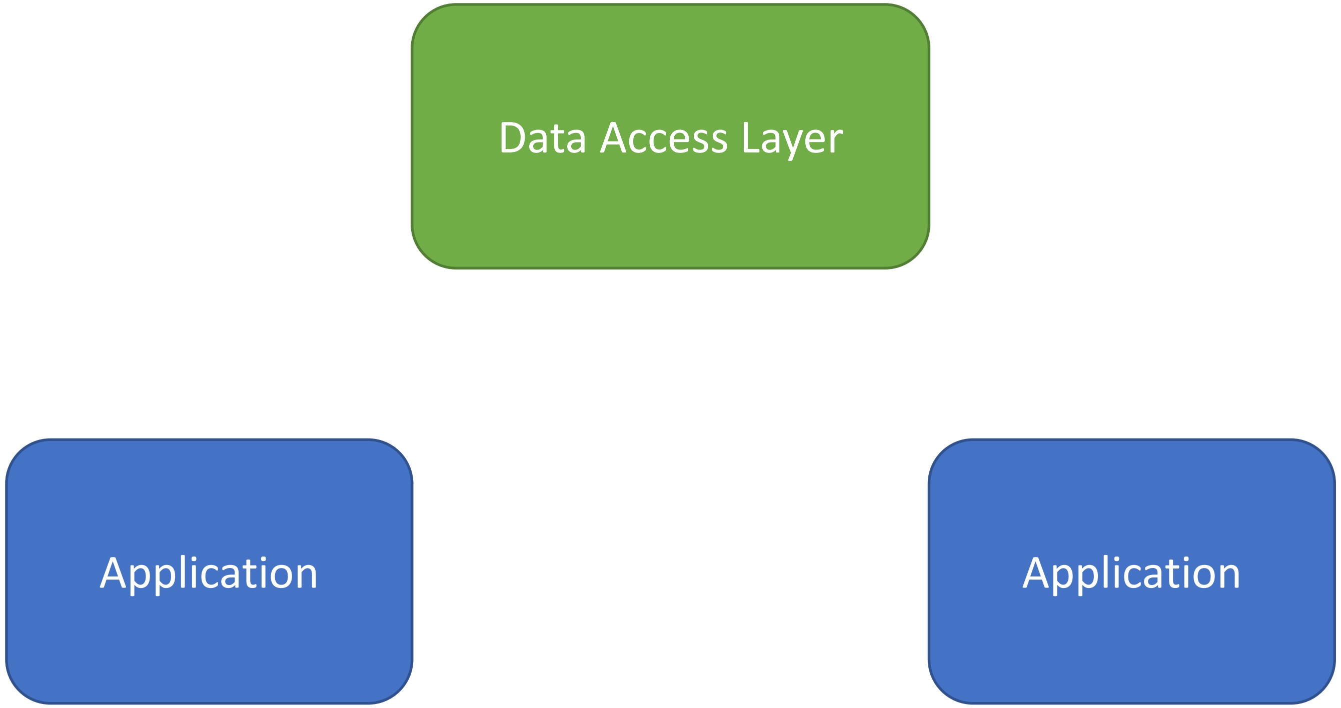


Data Access Layer

Application

Application

Pain Points

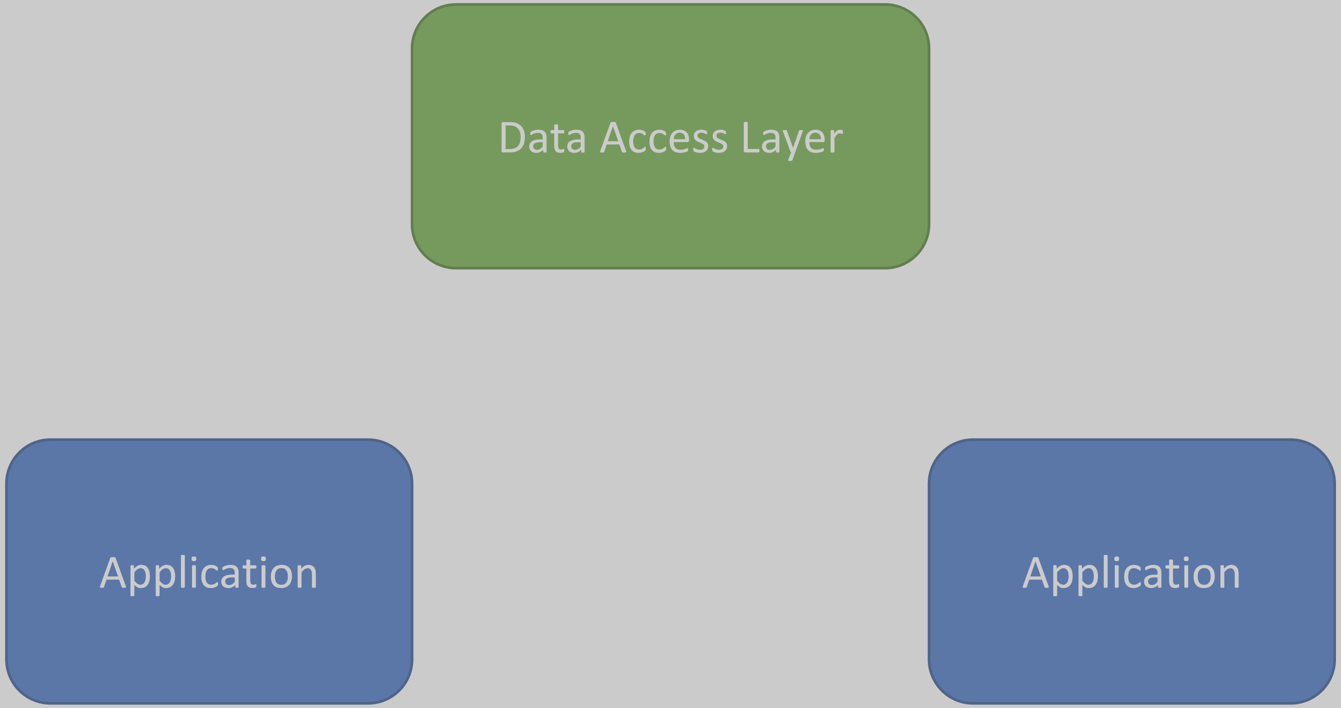


Data Access Layer

Application

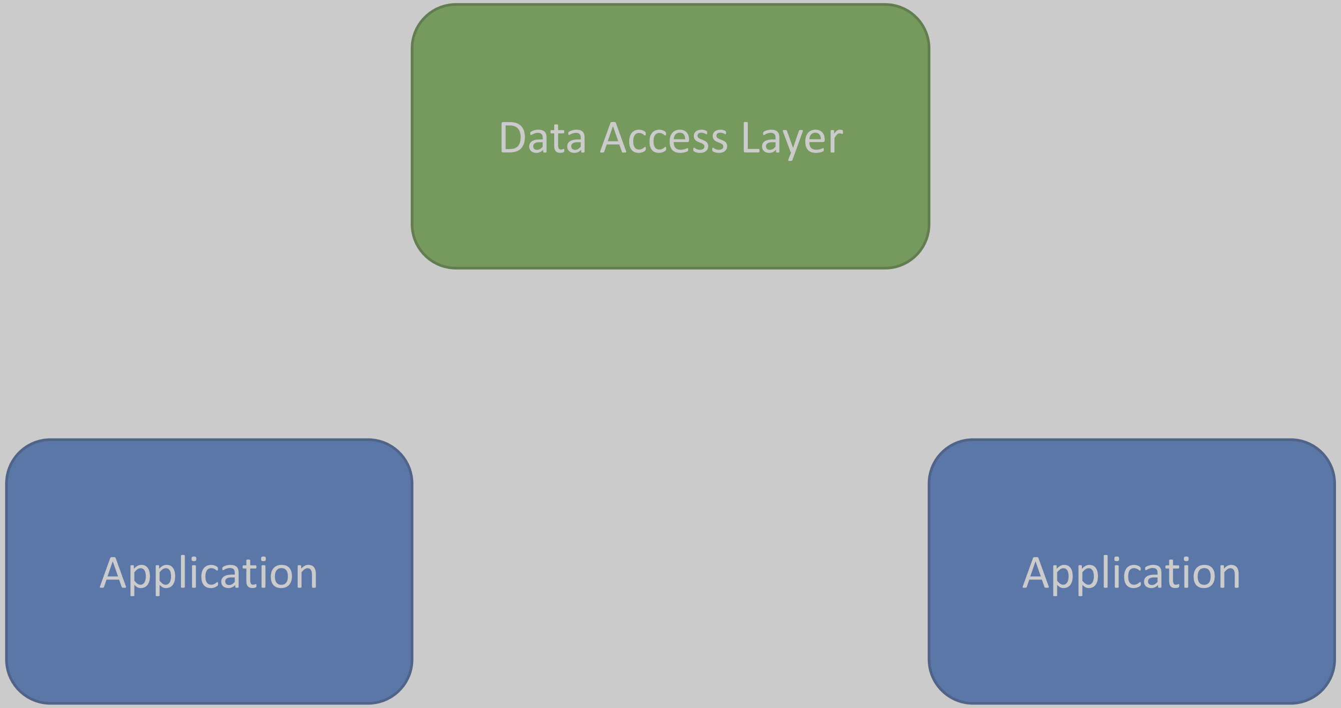
Application

Pain Points



1. Code change contention

Pain Points



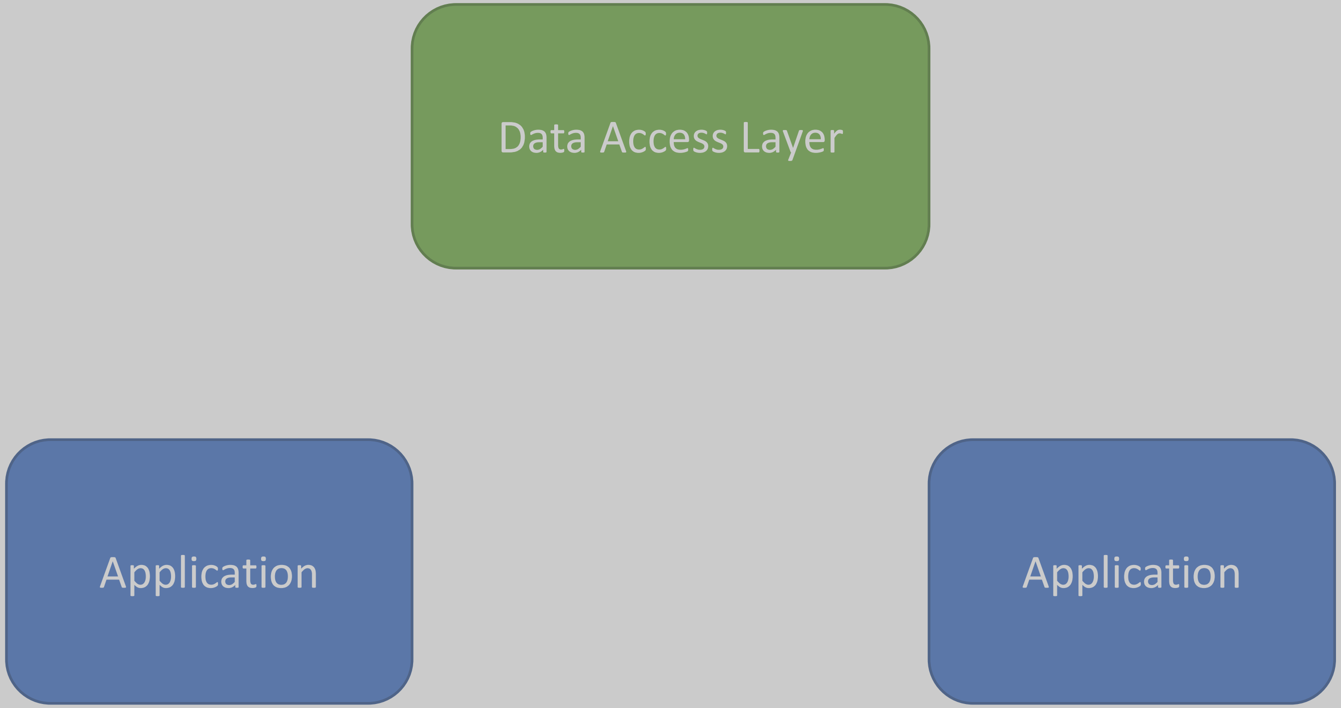
Data Access Layer

Application

Application

1. Code change contention
2. Performance differences

Pain Points



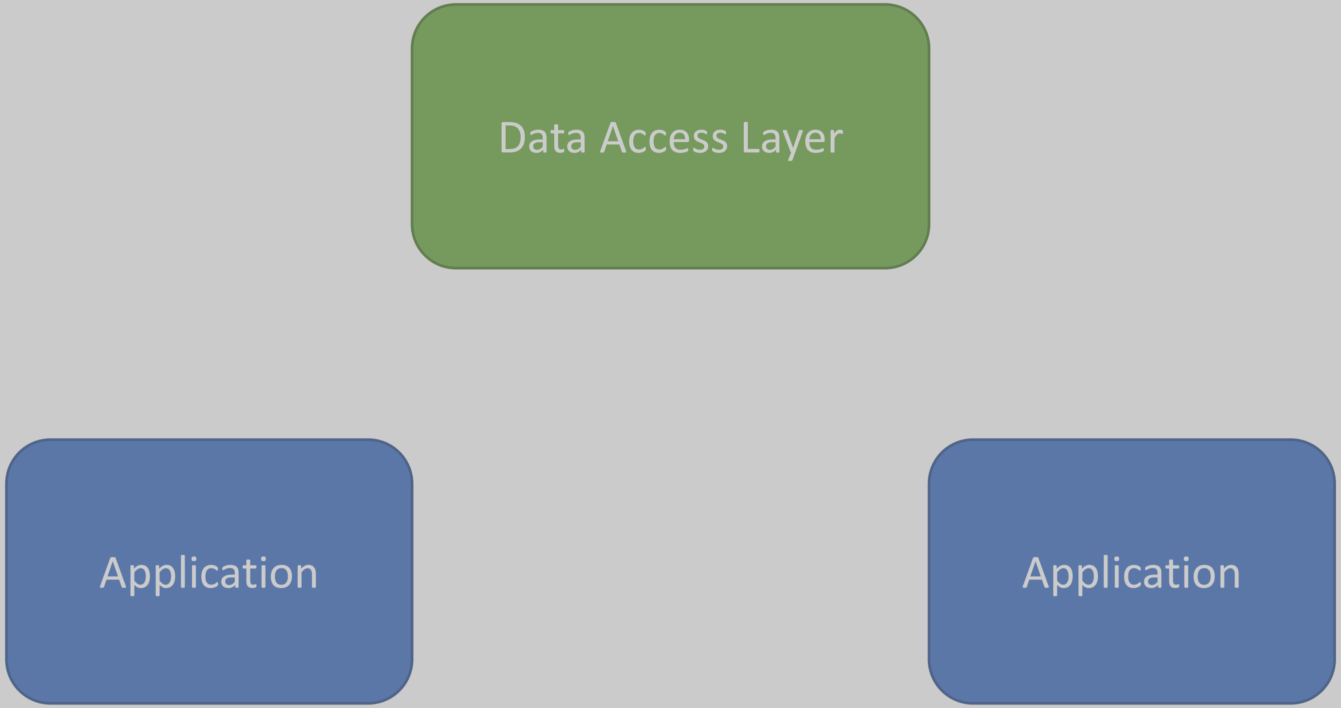
Data Access Layer

Application

Application

1. Code change contention
2. Performance differences
3. Unclear schema ownership

Pain Points



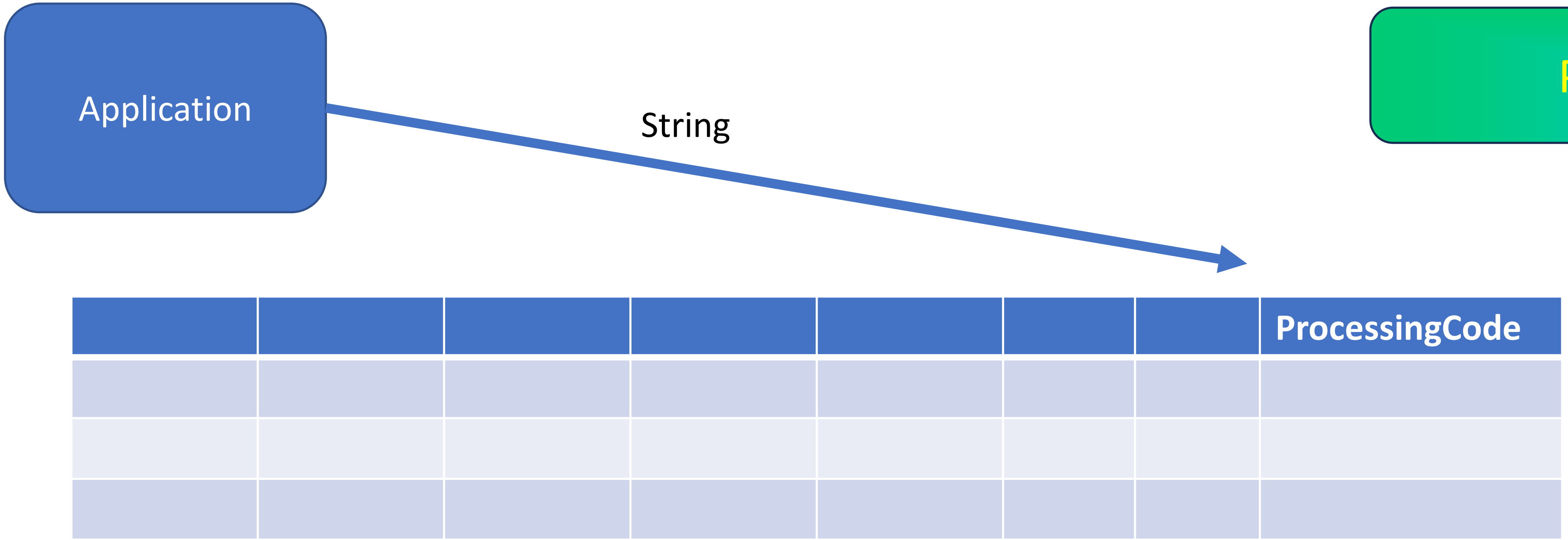
Data Access Layer

Application

Application

1. Code change contention
2. Performance differences
3. Unclear schema ownership
4. May need to consider duplicating code to allow apps to evolve independently

Pain Points

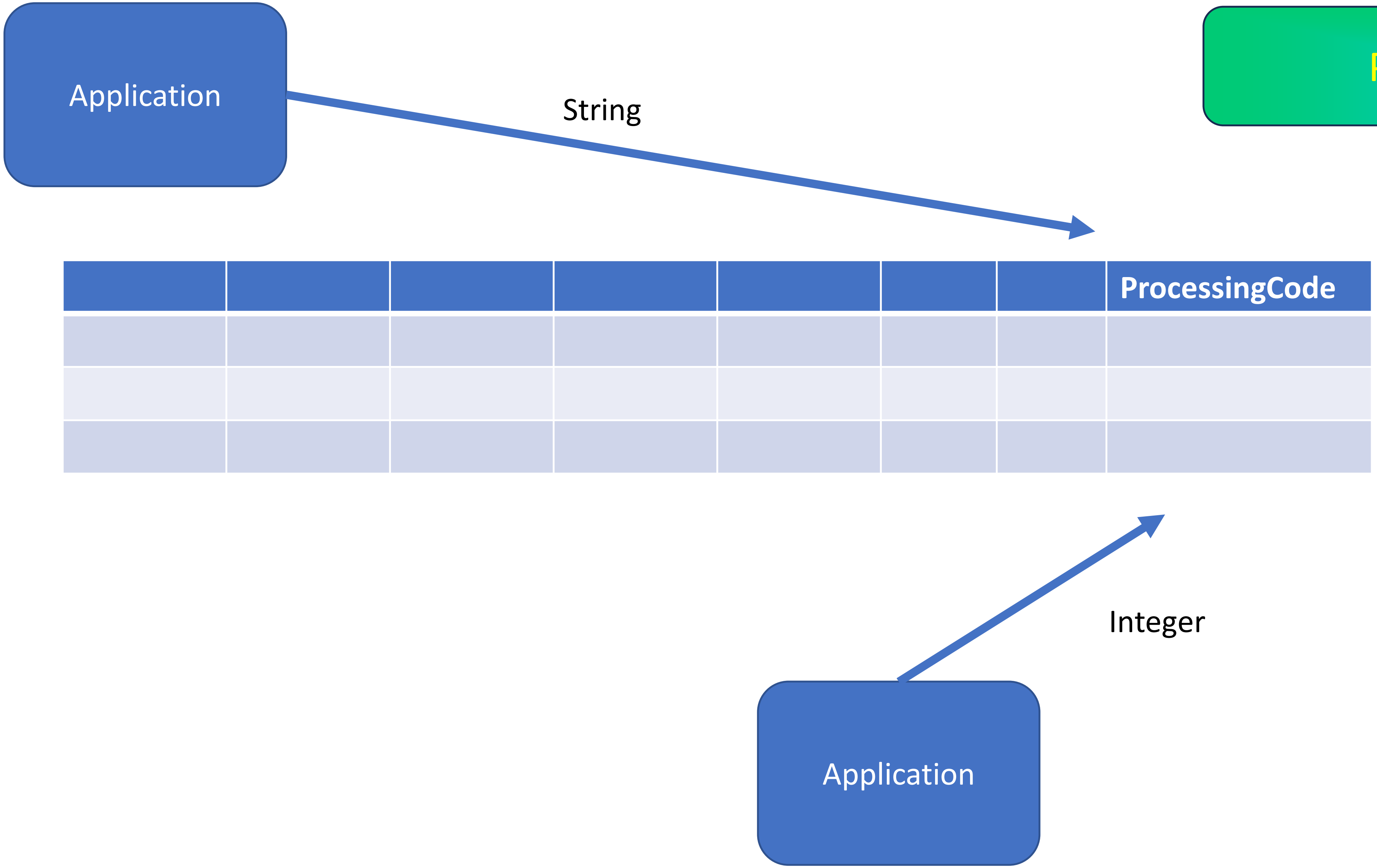


Application

String

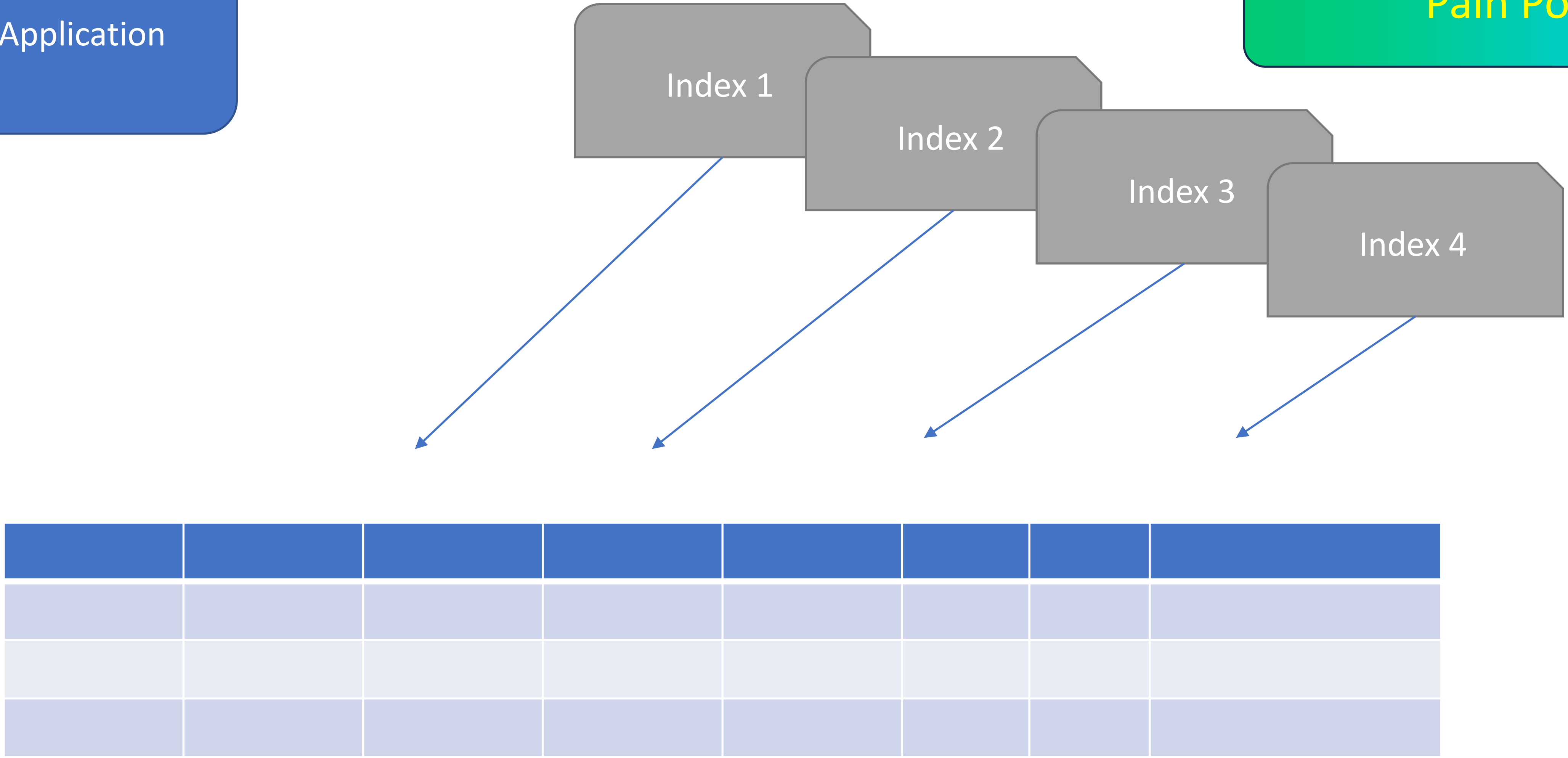
							ProcessingCode

Pain Points

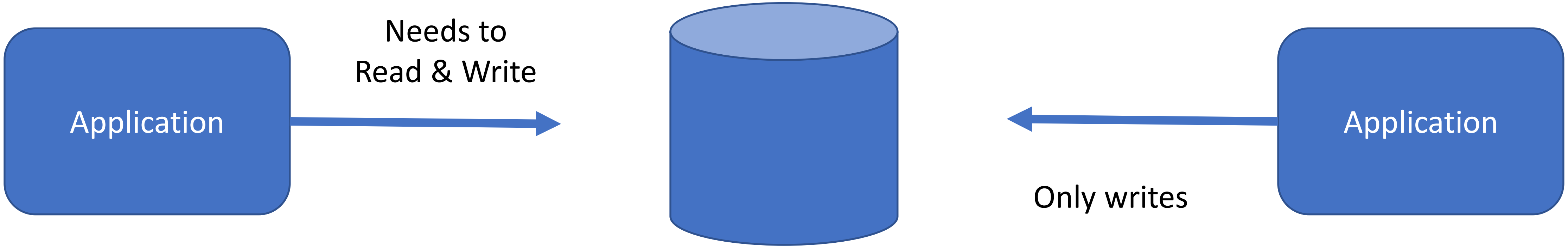


Pain Points

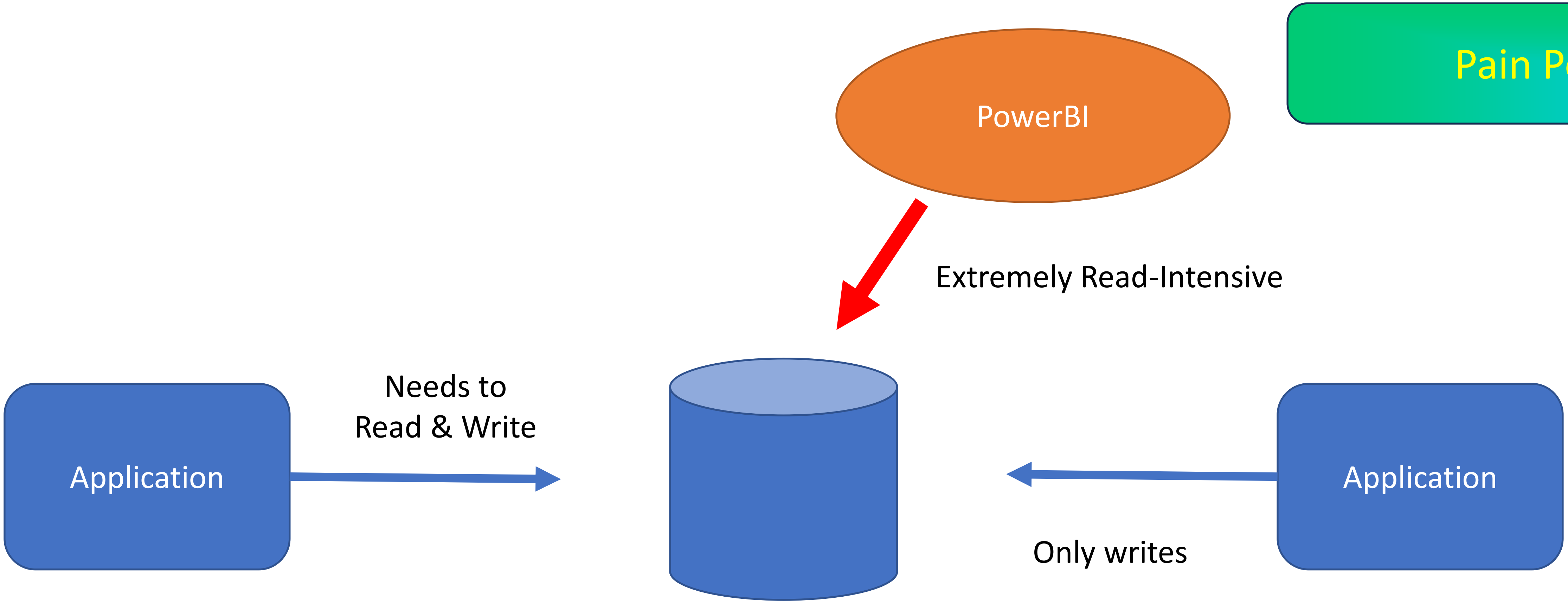
Application



Pain Points



Pain Points



PowerBI

Extremely Read-Intensive

Application

Needs to
Read & Write

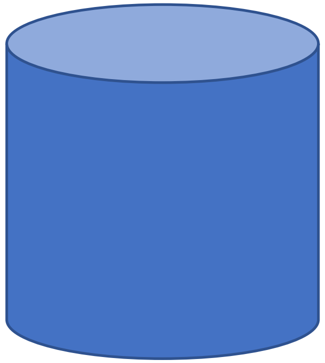
Only writes

Application

What is this “Data Store” of which you speak?

A database is a data store. But a data store is not necessarily a database.

A database is a data store. But a data store is not necessarily a database.



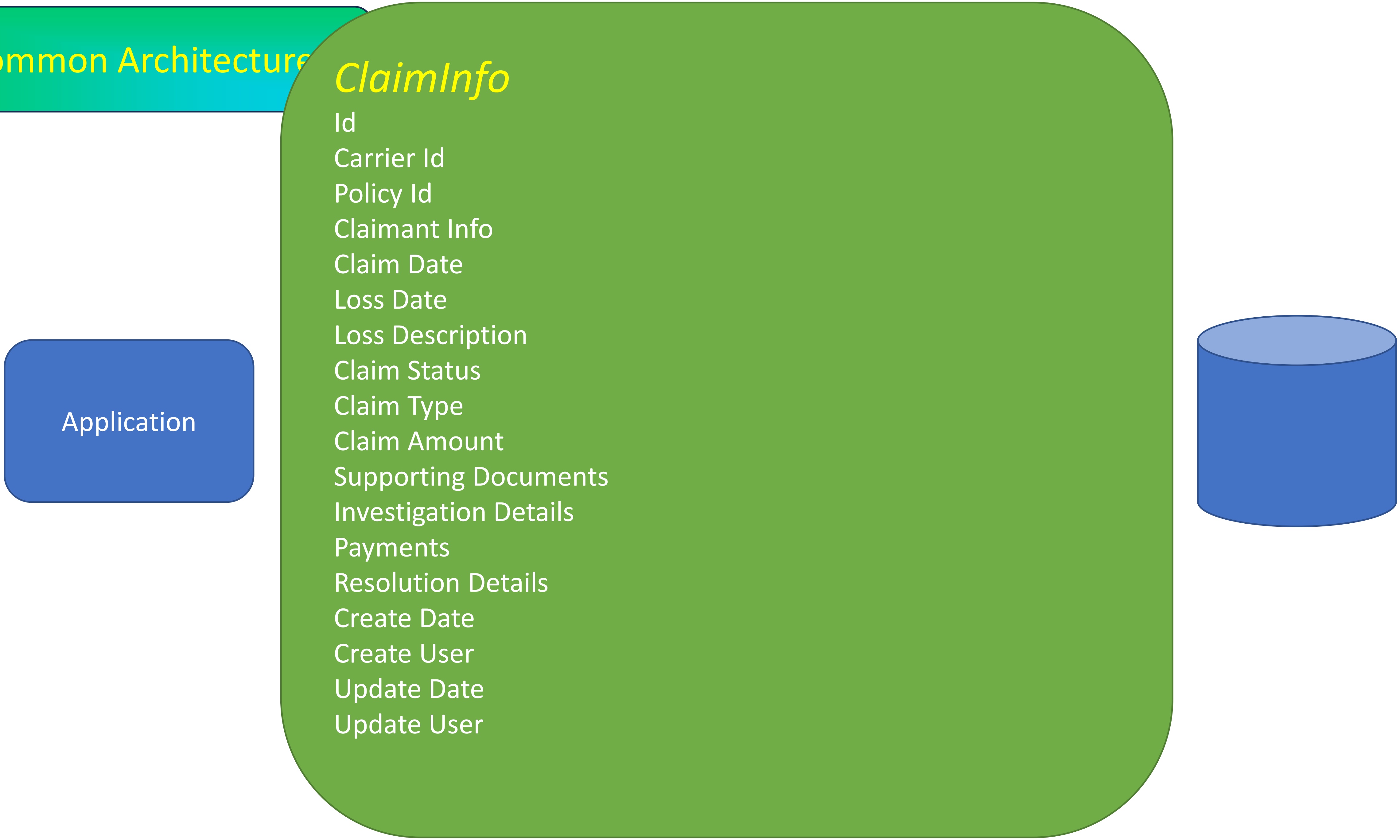
A database has an “engine” that maintains data integrity among many other functions.

A database is a data store. But a data store is not necessarily a database.

A database has an “engine” that maintains data integrity among many other functions.

A data store could be a database but could also be a File, Excel File, Azure File Share, Azure Table Storage, Azure Blob Account, AWS S3 Bucket, …….

Common Architecture



Application

ClaimInfo

Id
Carrier Id
Policy Id
Claimant Info
Claim Date
Loss Date
Loss Description
Claim Status
Claim Type
Claim Amount
Supporting Documents
Investigation Details
Payments
Resolution Details
Create Date
Create User
Update Date
Update User

ClaimInfoDetails

Id

Carrier Id

Policy Id

Claimant Info

Claim Date

Loss Date

Loss Description

Claim Status

Claim Type

Claim Amount

Supporting Documents

Investigation Details

Payments

Resolution Details

Create Date

Create User

Update Date

Update User

NewClaimInfo

Id
Carrier Id
Policy Id
Claimant Info
Claim Date
Loss Date
Loss Description
Claim Status
Claim Type
Claim Amount
~~Supporting Documents~~
~~Investigation Details~~
~~Payments~~
~~Resolution Details~~
Create Date
Create User
~~Update Date~~
~~Update User~~

NewClaimInfo

Id
Carrier Id
Policy Id
Claimant Info
Claim Date
Loss Date
Loss Description
Claim Status
Claim Type
Claim Amount
Create Date
Create User

AddClaimInfoDocs

Id
Policy Id
Supporting Documents
Create Date
Create User

AddClaimPayment

Id
Policy Id
Payment Detail
Payment Amount
Create Date
Create User

AmendClaimInvestigationDetail

Claim Id
Investigation Id
Investigation Detail

Write Models

NewClaimInfo

Id
Carrier Id
Policy Id
Claimant Info
Claim Date
Loss Date
Loss Description
Claim Status
Create Date
Create User

AddClaimInfoDocs

Id
Policy Id
Supporting Documents
Create Date
Create User

AmendClaimInvestigationDetail

Claim Id
Investigation Id
Investigation Detail

AddClaimP

Policy Id
Payment Detail
Payment Amount
Create Date
Create User

ClaimInfoDetails

Id
Carrier Id
Policy Id
Claimant Info
Claim Date
Loss Date
Loss Description
Claim Status
Claim Type
Claim Amount
Supporting Documents
Investigation Details
Payments
Resolution Details
Create Date
Create User
Update Date
Update User

ClaimInfoDetails

Id
Carrier Id
Policy Id
Claimant Info
Claim Date
Loss Date
Loss Description
Claim Status
Claim Type
Claim Amount
Supporting Documents
Investigation Details
Payments
Resolution Details
Create Date
Create User
Update Date
Update User

ClaimSupportDocs

Id
Carrier Id
Policy Id
Supporting Documents

ClaimAppliedPmts

Id
Carrier Id
Policy Id
Payment Details
Payment Amount

ClaimStatusHistory

Id
Policy Id
Claim Status

ClaimResolutionDetails

Id
Policy Id
ResolutionDetails

ClaimInvestigationDetails

Id
Policy Id
Investigation Details

Read Models

ClaimInfoDetails

Id
Carrier Id
Policy Id
Claimant Info
Claim Date
Loss Date
Loss Description
Claim Status

Supporting Documents
Investigation Details
Payments
Resolution Details
Create Date
Create User
Update Date
Update User

ClaimSupportDocs

Id
Carrier Id
Policy Id
Supporting Documents

ClaimAppliedPmts

Policy Id
Payment Details
Payment Amount

ClaimStatusHistory

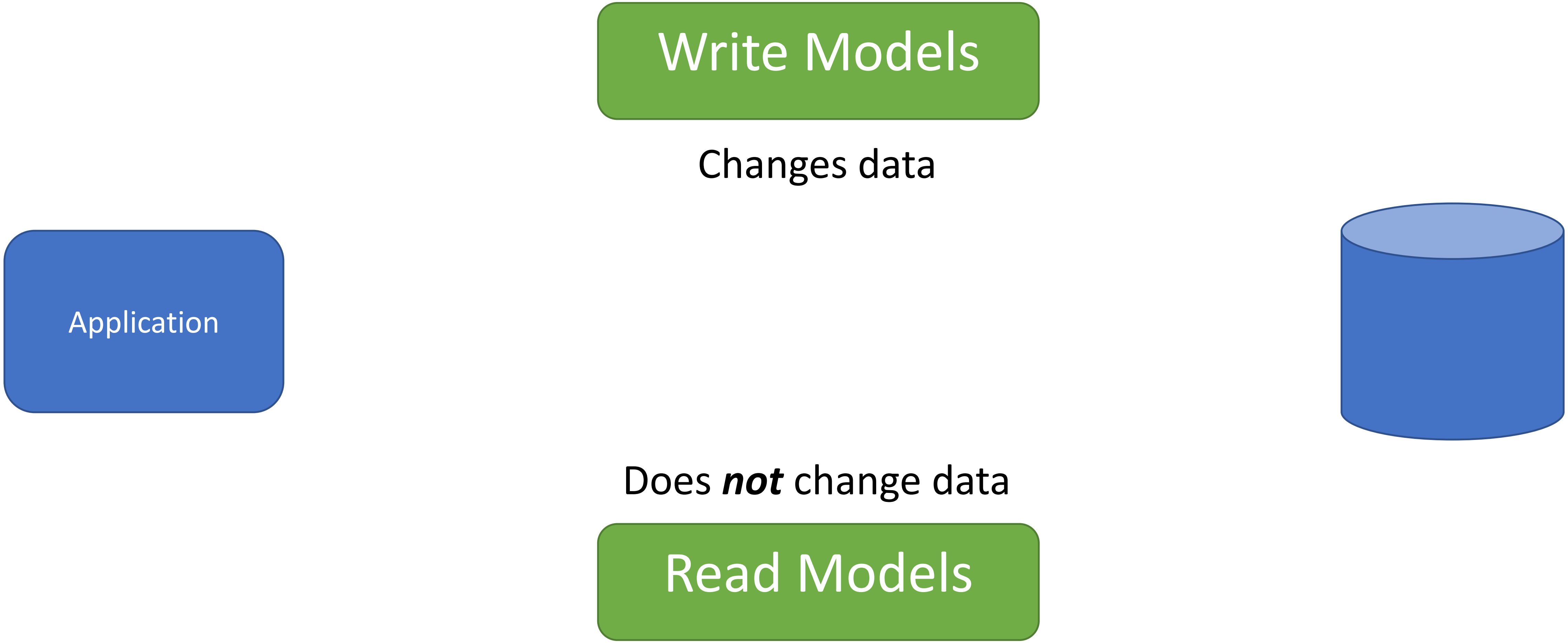
Id
Policy Id
Claim Status

ClaimResolutionDetails

Id
Policy Id

ClaimInvestigationDetails

Id
Policy Id
Investigation Details



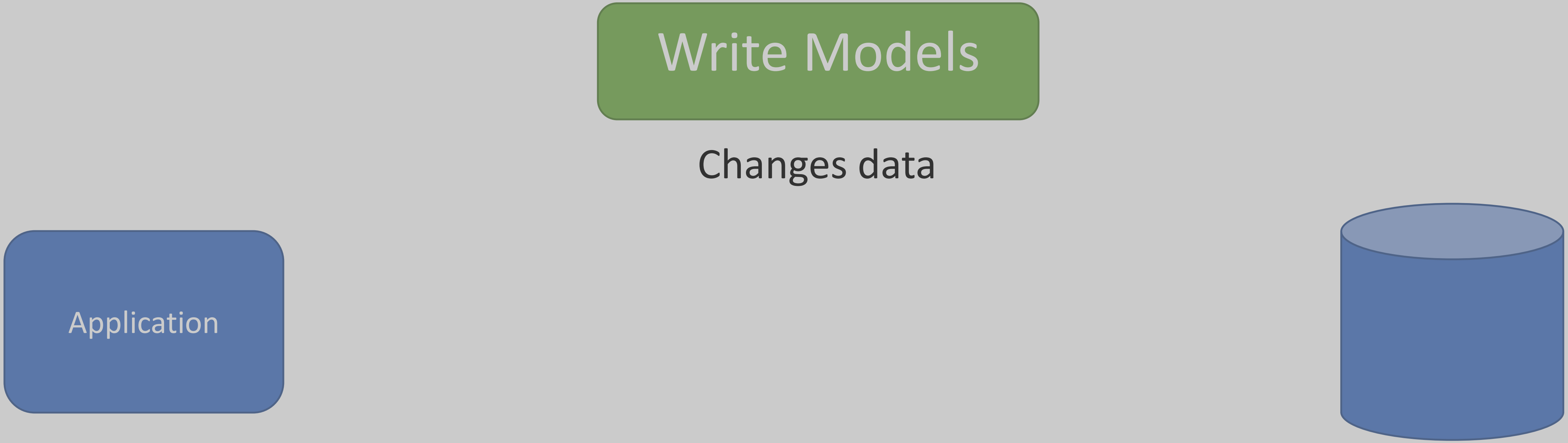
Write Models

Changes data

Application

Does ***not*** change data

Read Models



Write Models

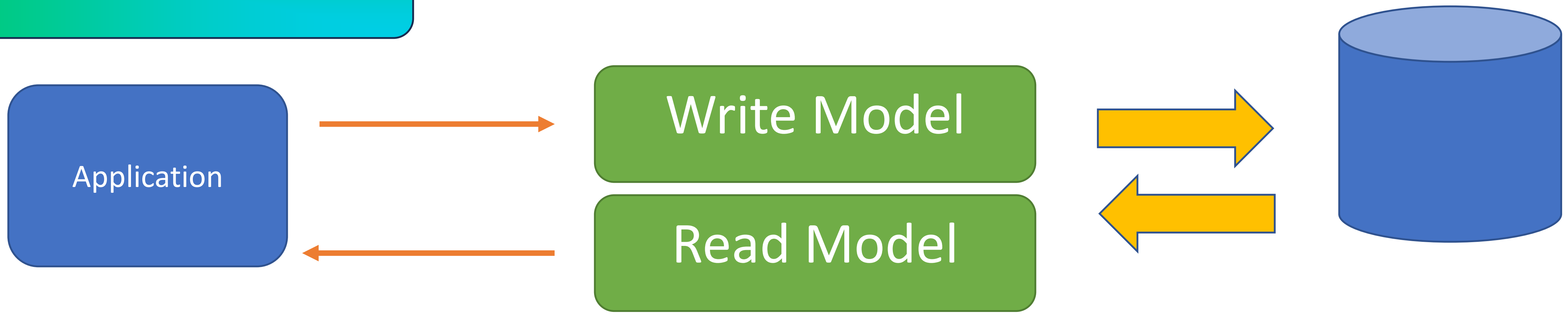
Changes data

Application

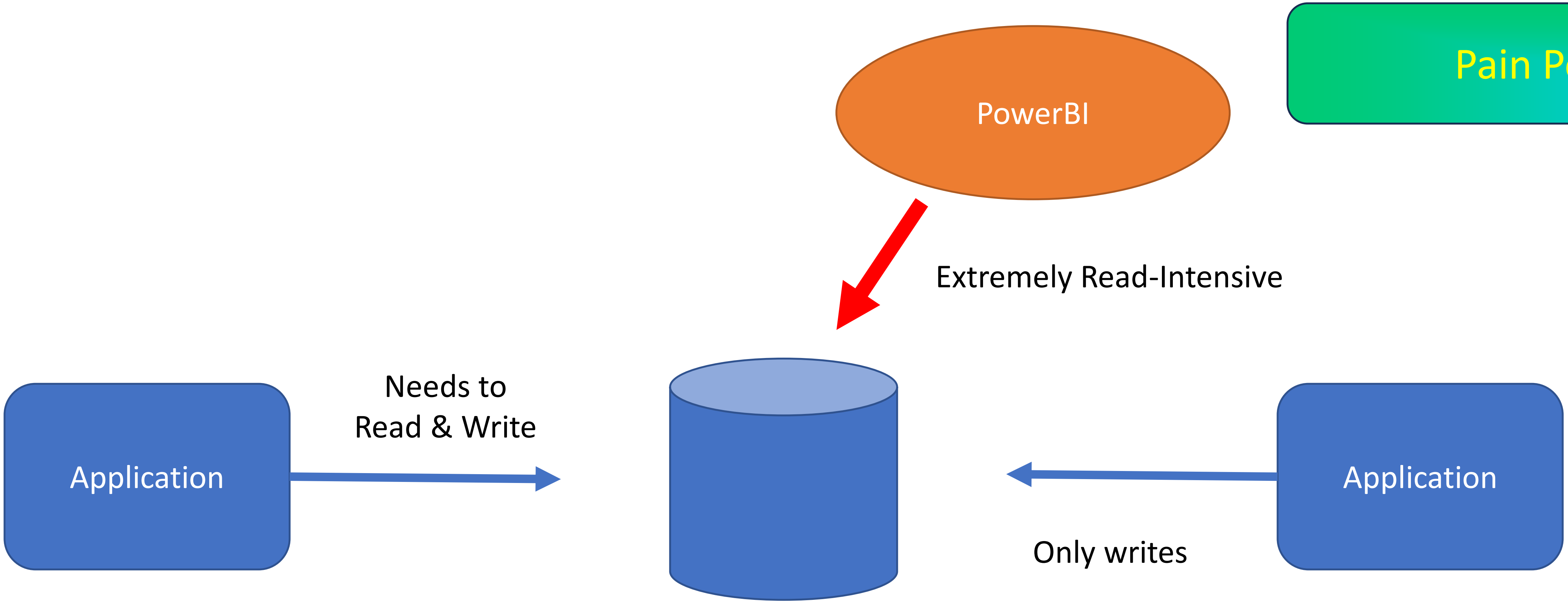
CQRS

Command – Query – Responsibility - Segregation

CQRS



Pain Points



PowerBI

Extremely Read-Intensive

Application

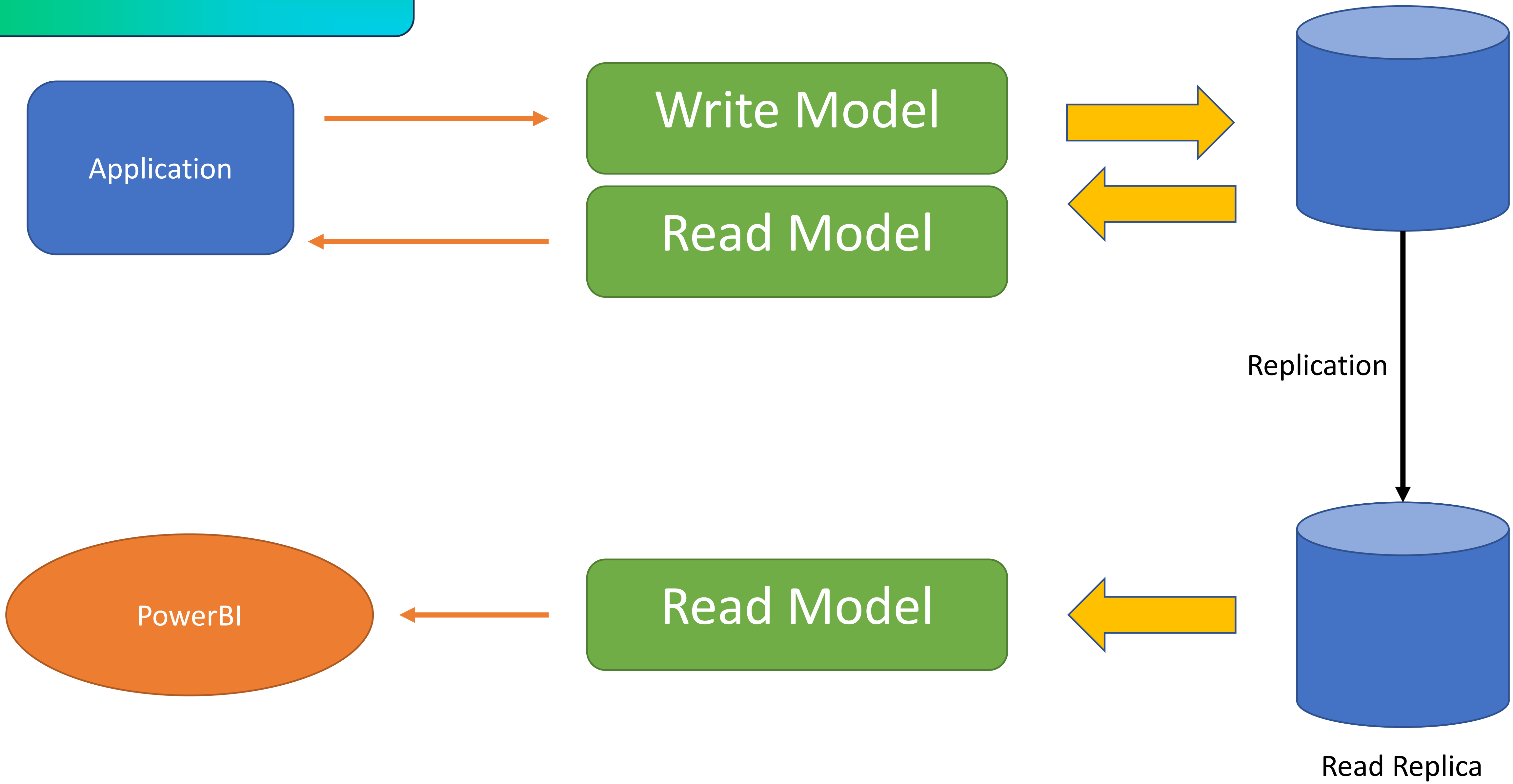
Needs to
Read & Write

Only writes

Application

CQRS

*Does not have to be Write-Only



Application

Write Model

Read Model

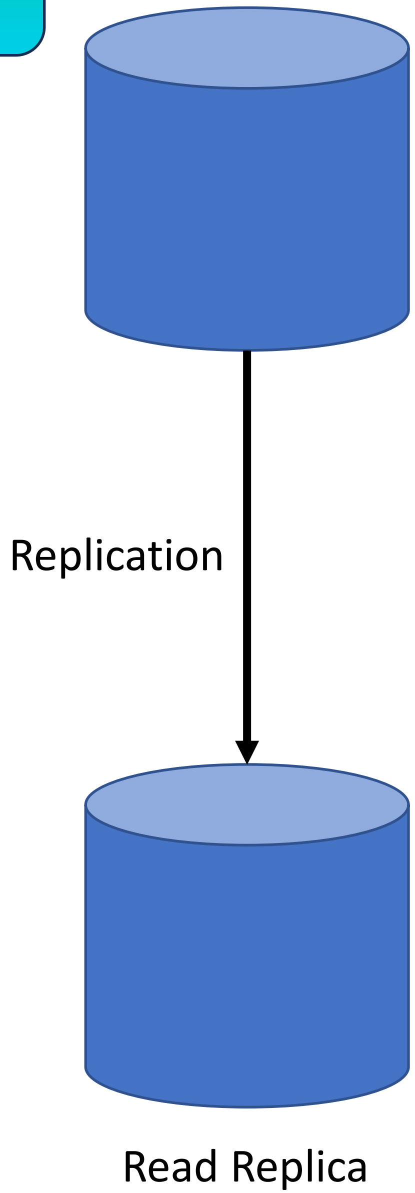
Replication

PowerBI

Read Model

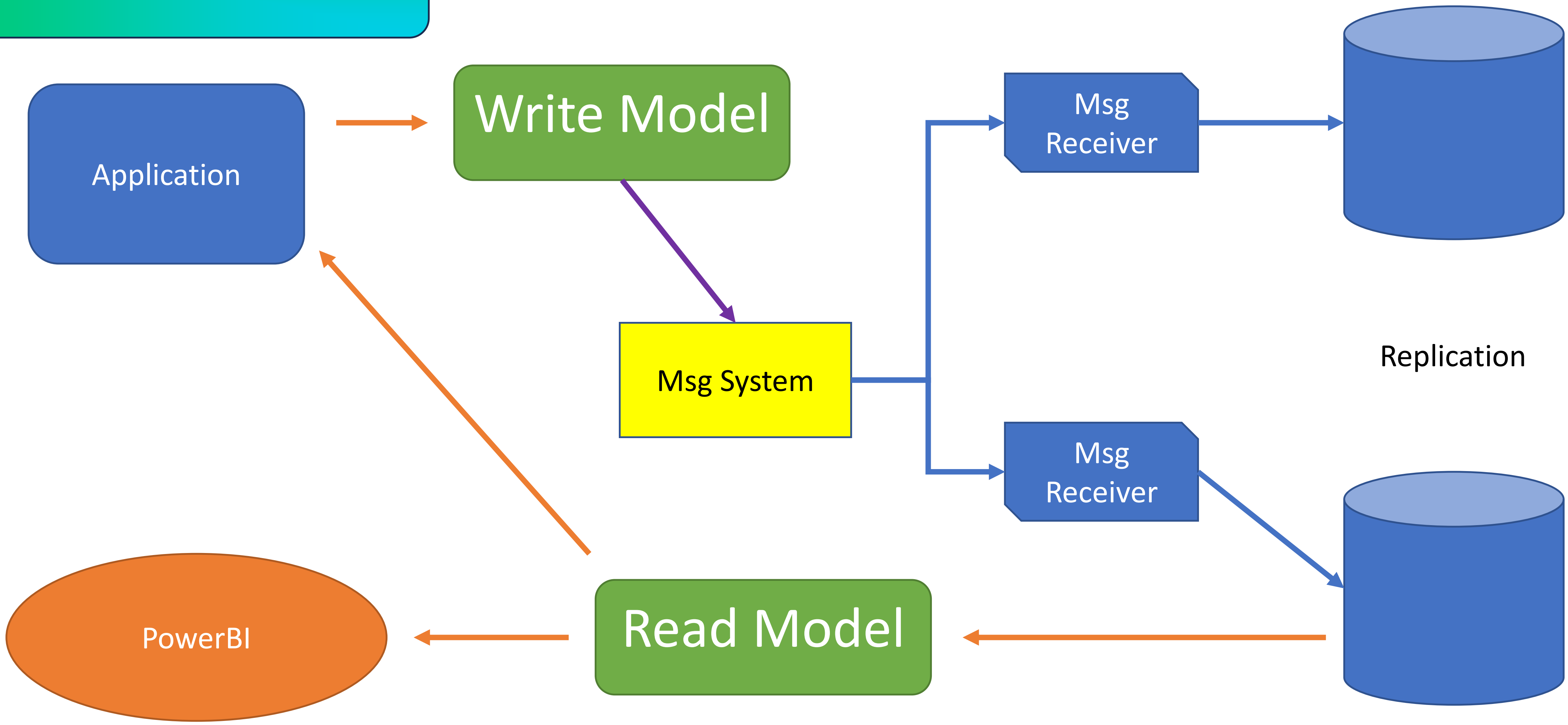
Read Replica

Replication



MySQL Replication
Postgres – Streaming Replication
SQL Server – Transactional Replication
MongoDB – Replication Sets
Amazon Aurora

CQRS



Application

Write Model

Msg System

Msg Receiver

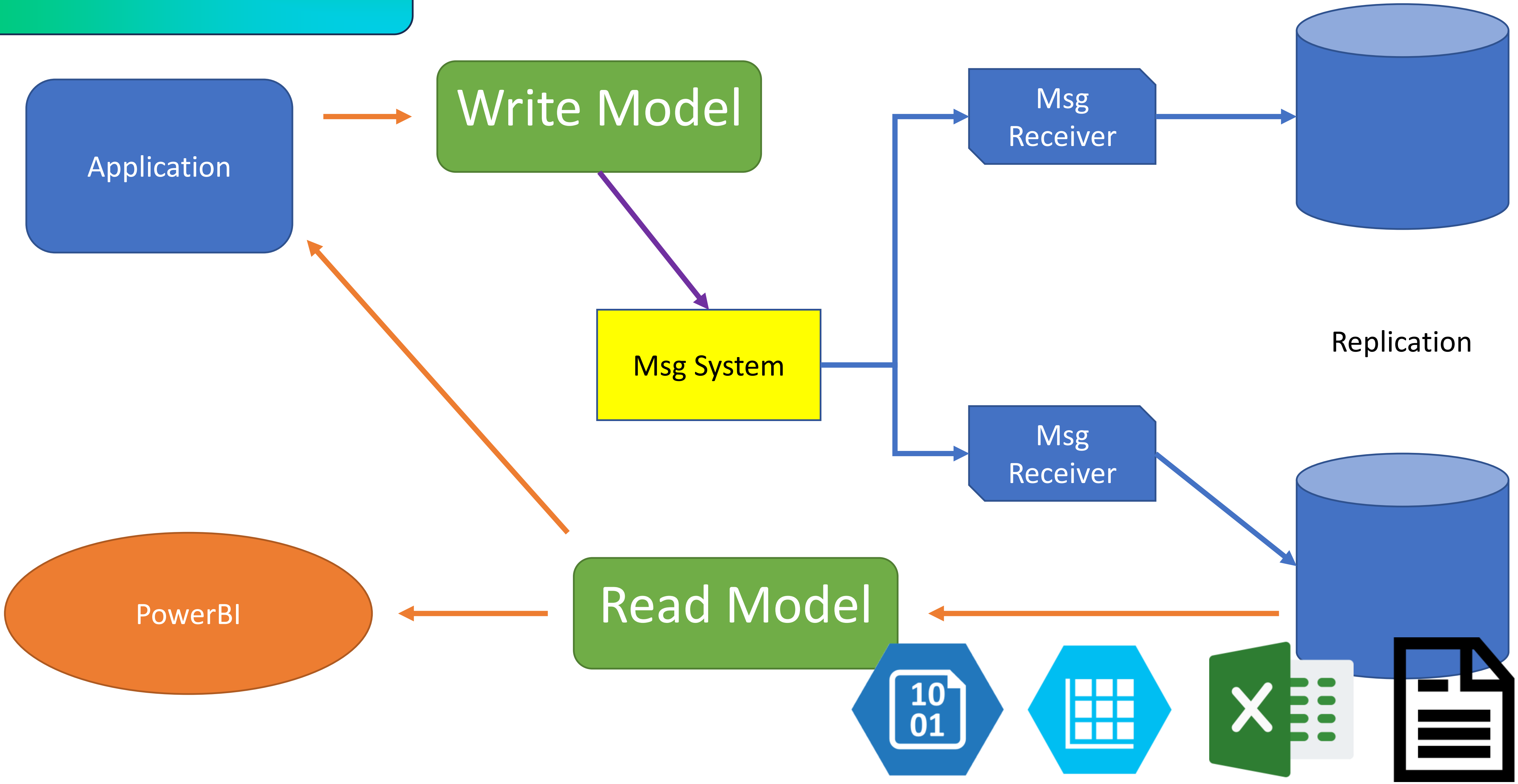
Msg Receiver

Replication

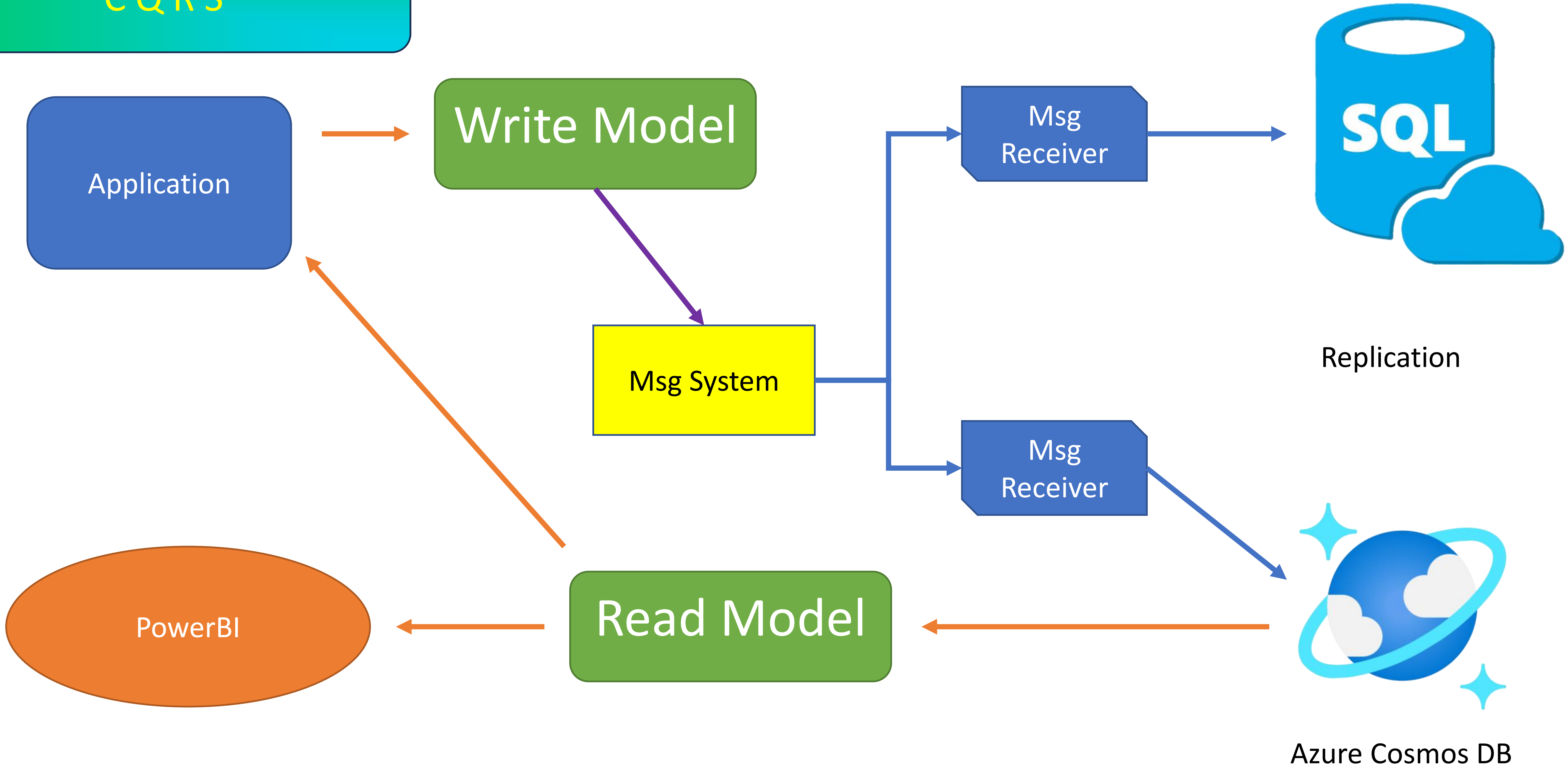
PowerBI

Read Model

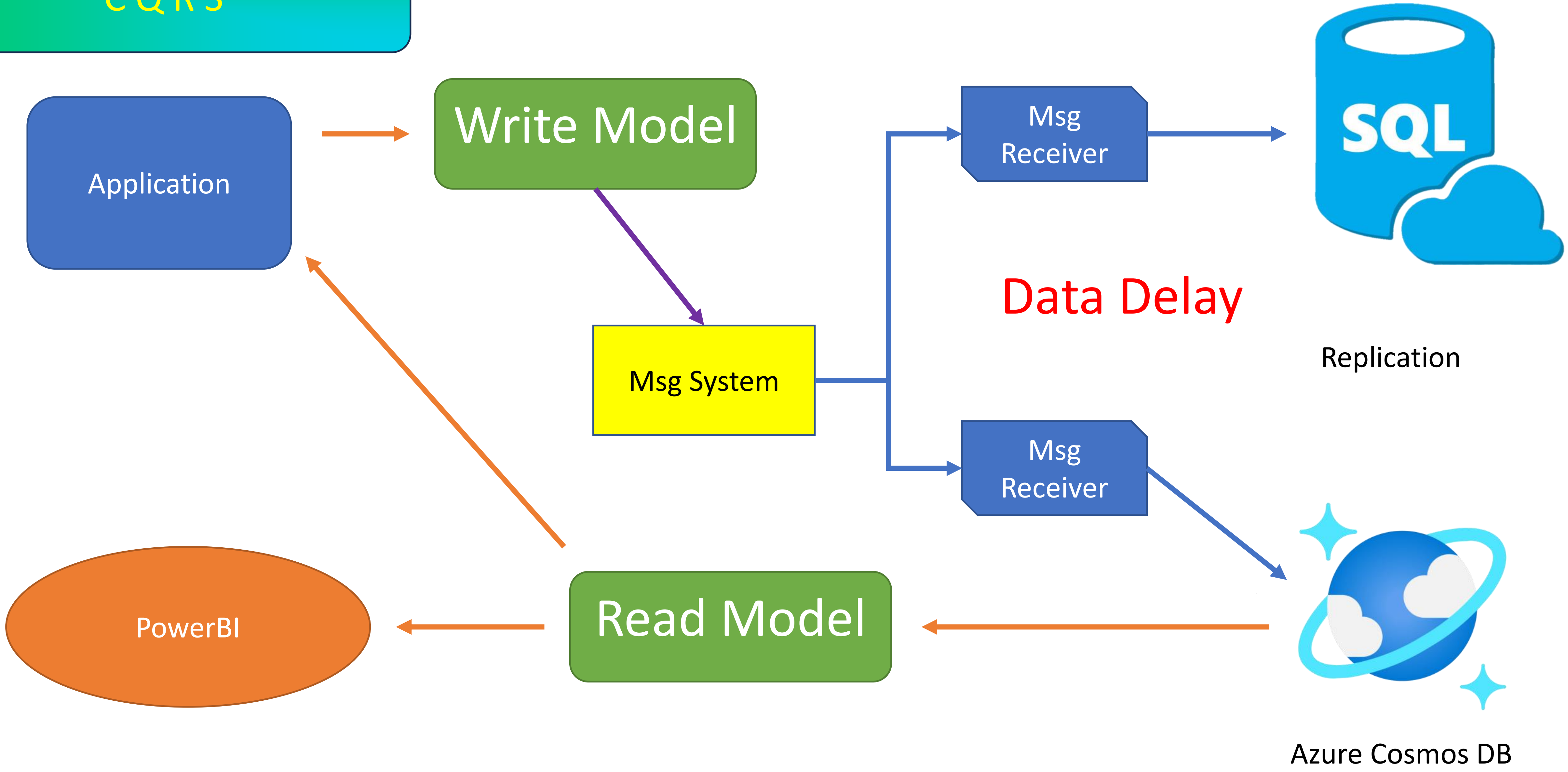
CQRS


Application
Write Model
Msg System
Msg Receiver
Msg Receiver
Replication
PowerBI
Read Model
10
01

CQRS


Application
Write Model
Msg System
Msg Receiver
SQL
Replication
Msg Receiver
Azure Cosmos DB
PowerBI
Read Model

CQRS



Application

Write Model

Msg System

Msg Receiver

Msg Receiver

SQL

Data Delay

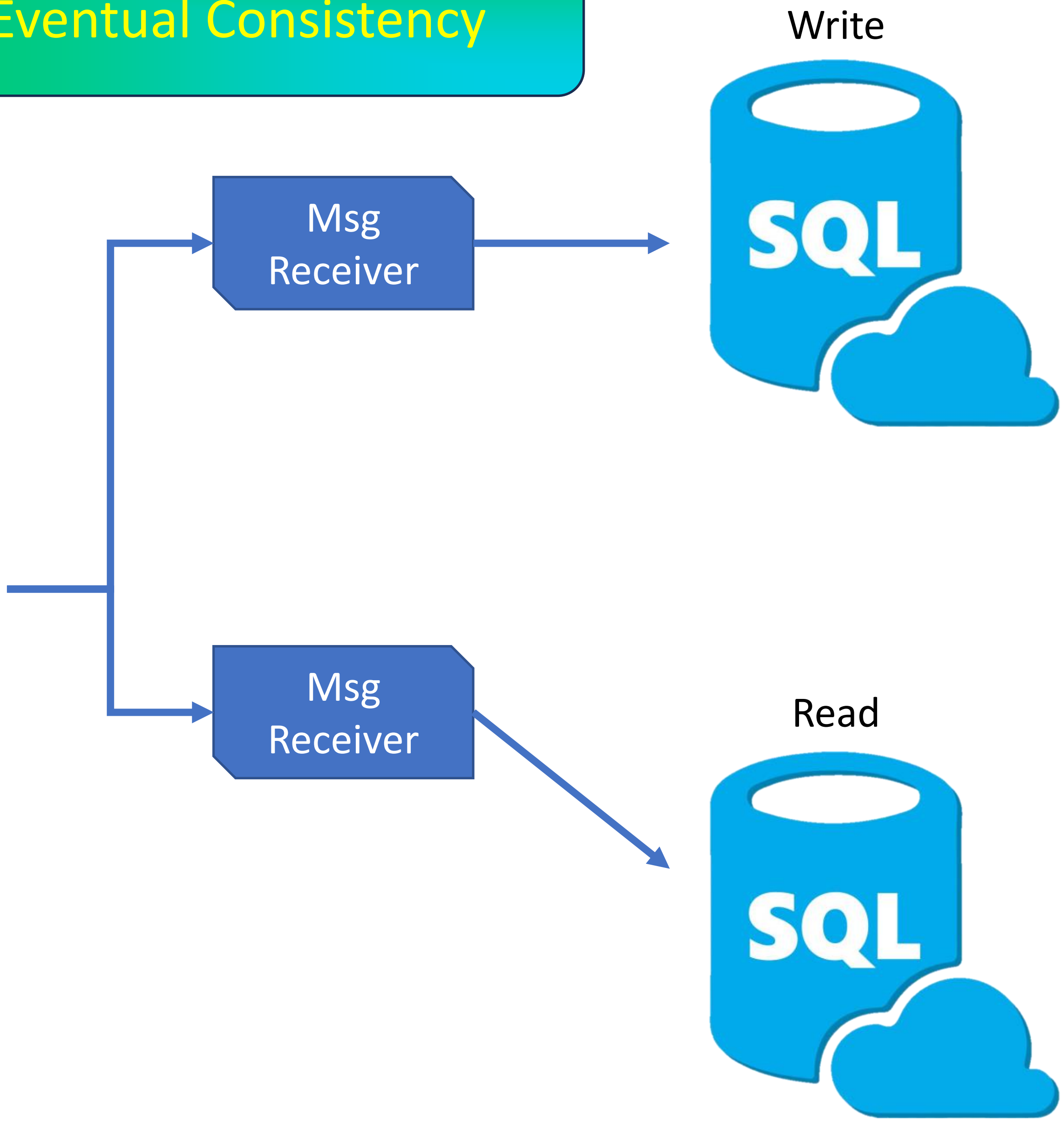
Replication

Read Model

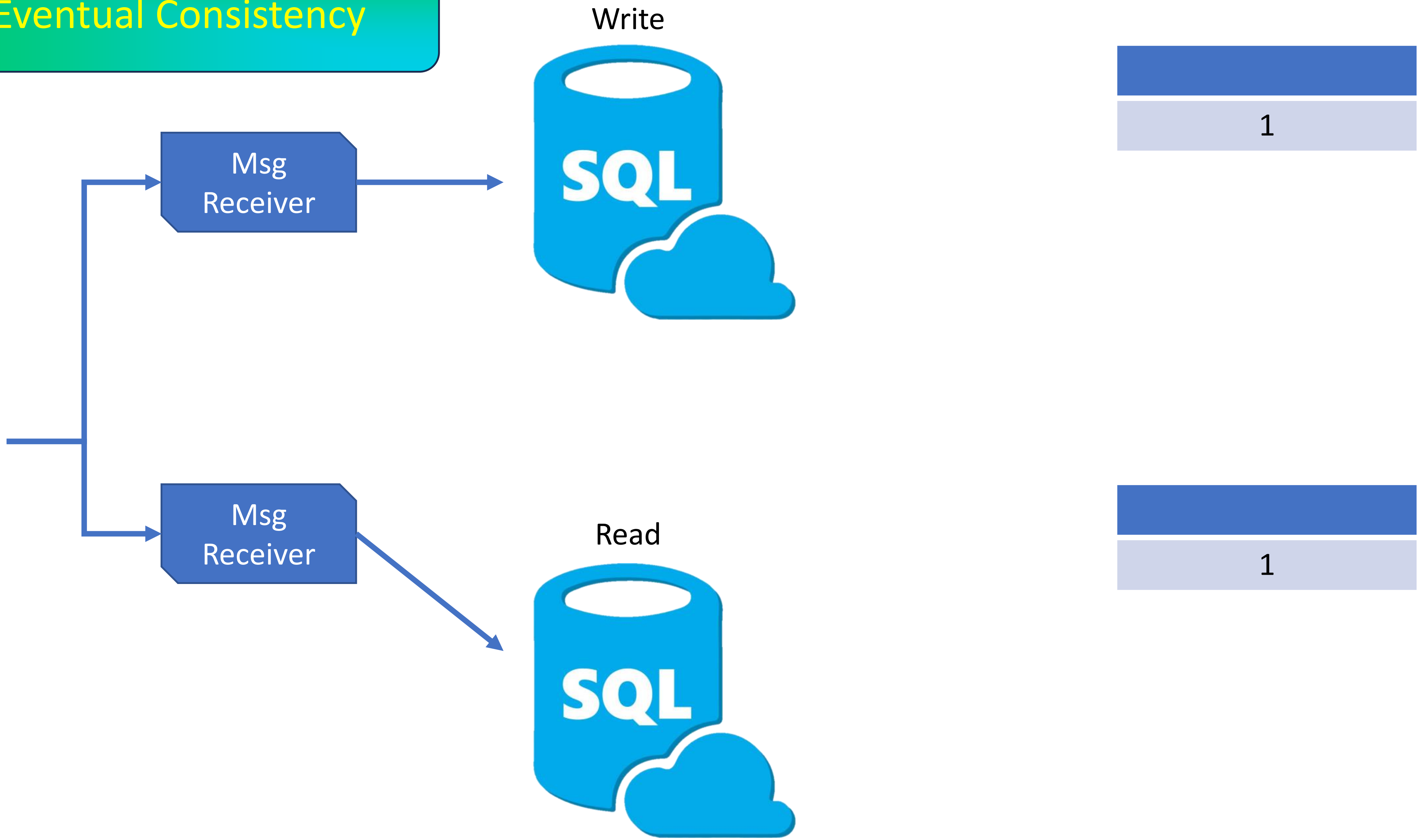
PowerBI

Azure Cosmos DB

Eventual Consistency



Eventual Consistency



Write

Msg Receiver

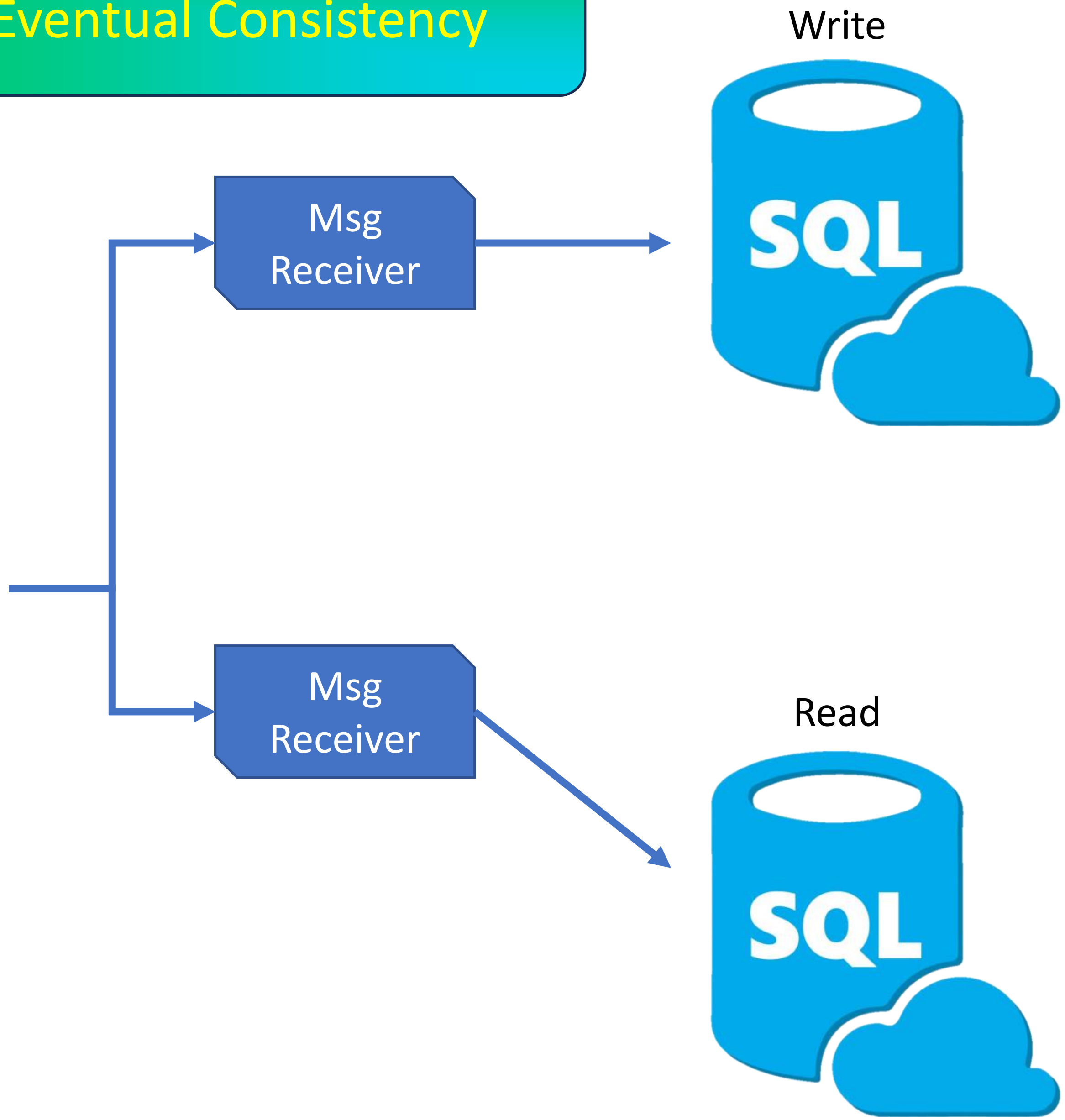
Msg Receiver

Read

1

1

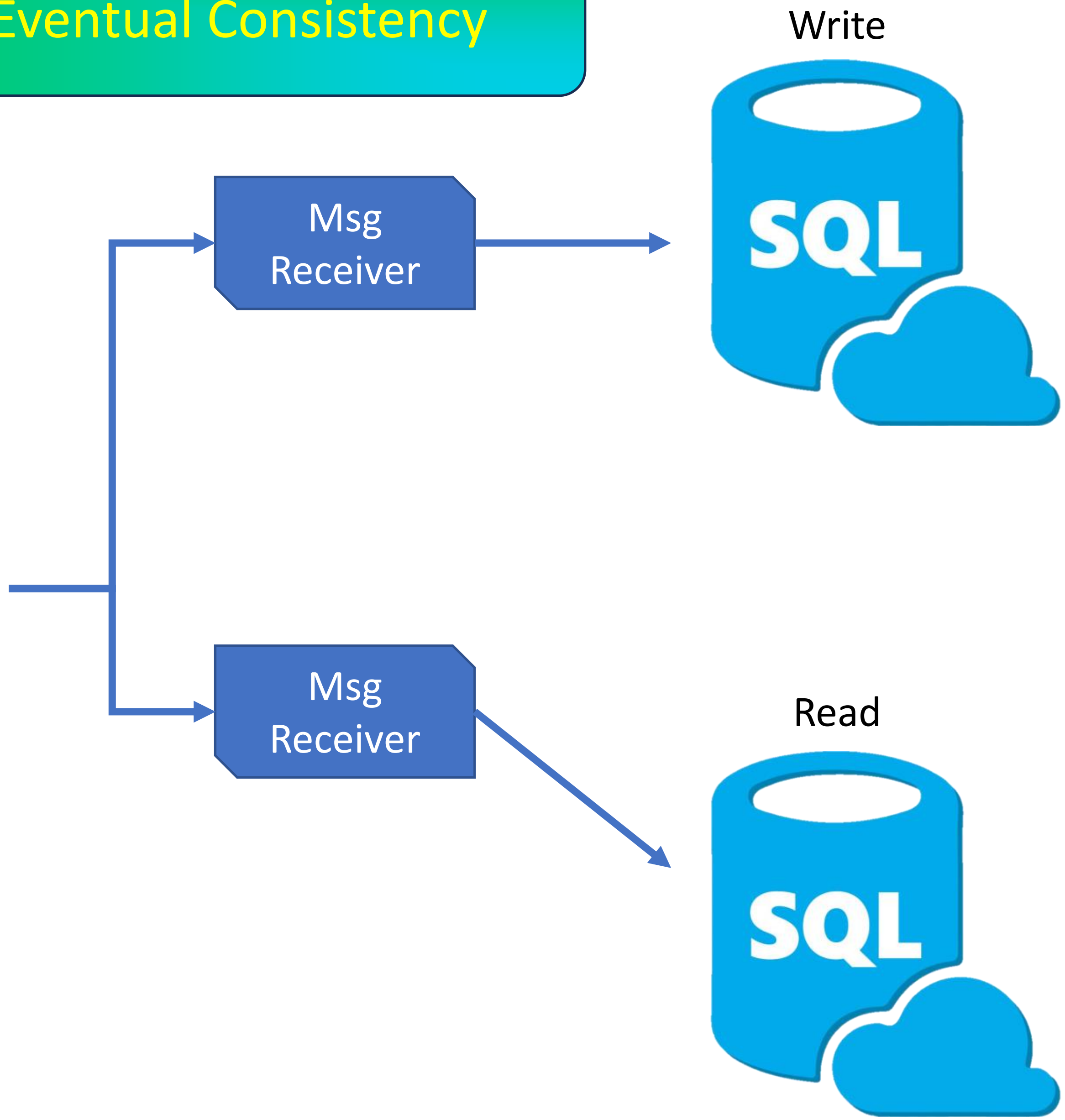
Eventual Consistency



1
2

1
2

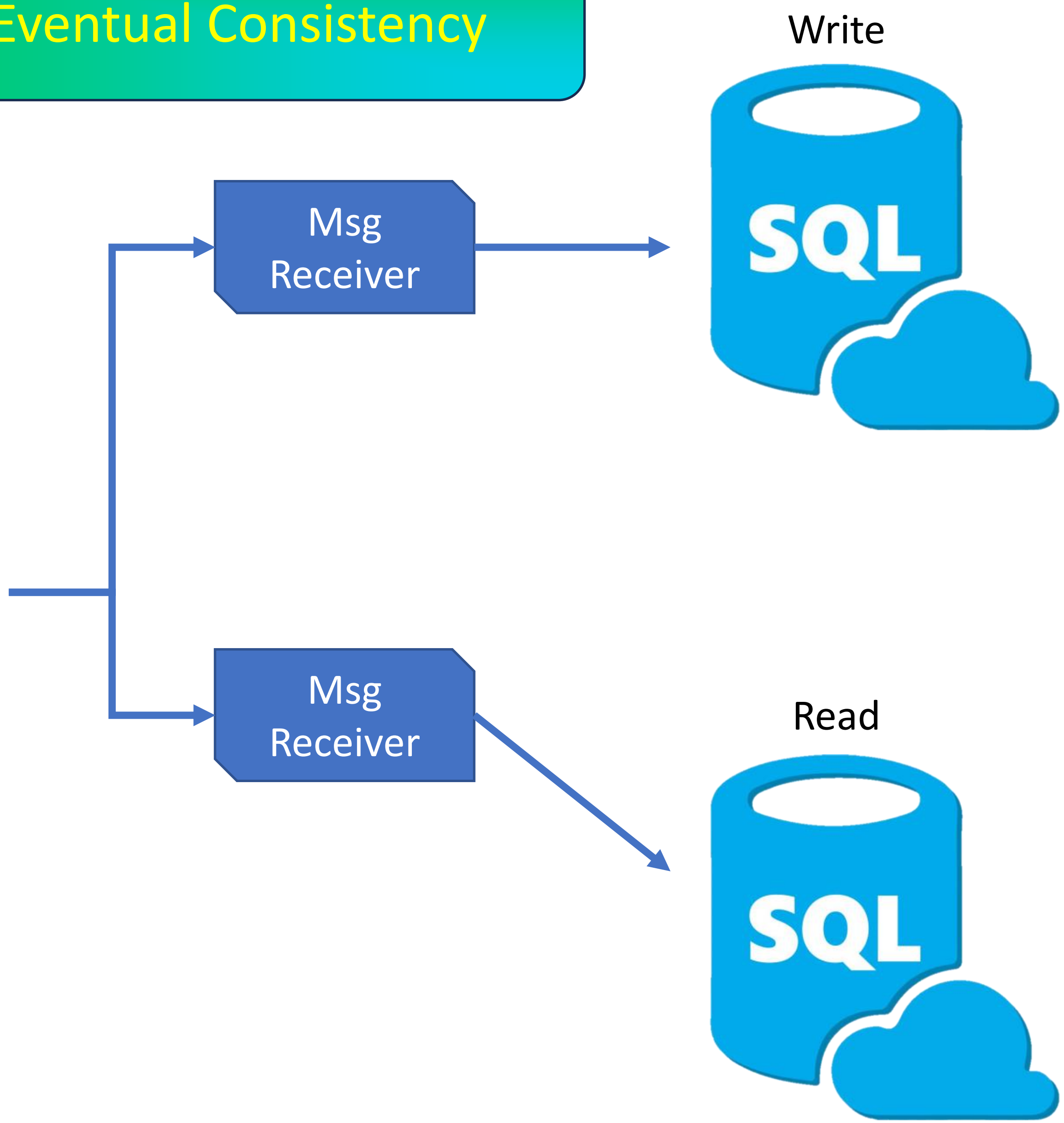
Eventual Consistency



1
2
3

1
2

Eventual Consistency



Write

Msg Receiver

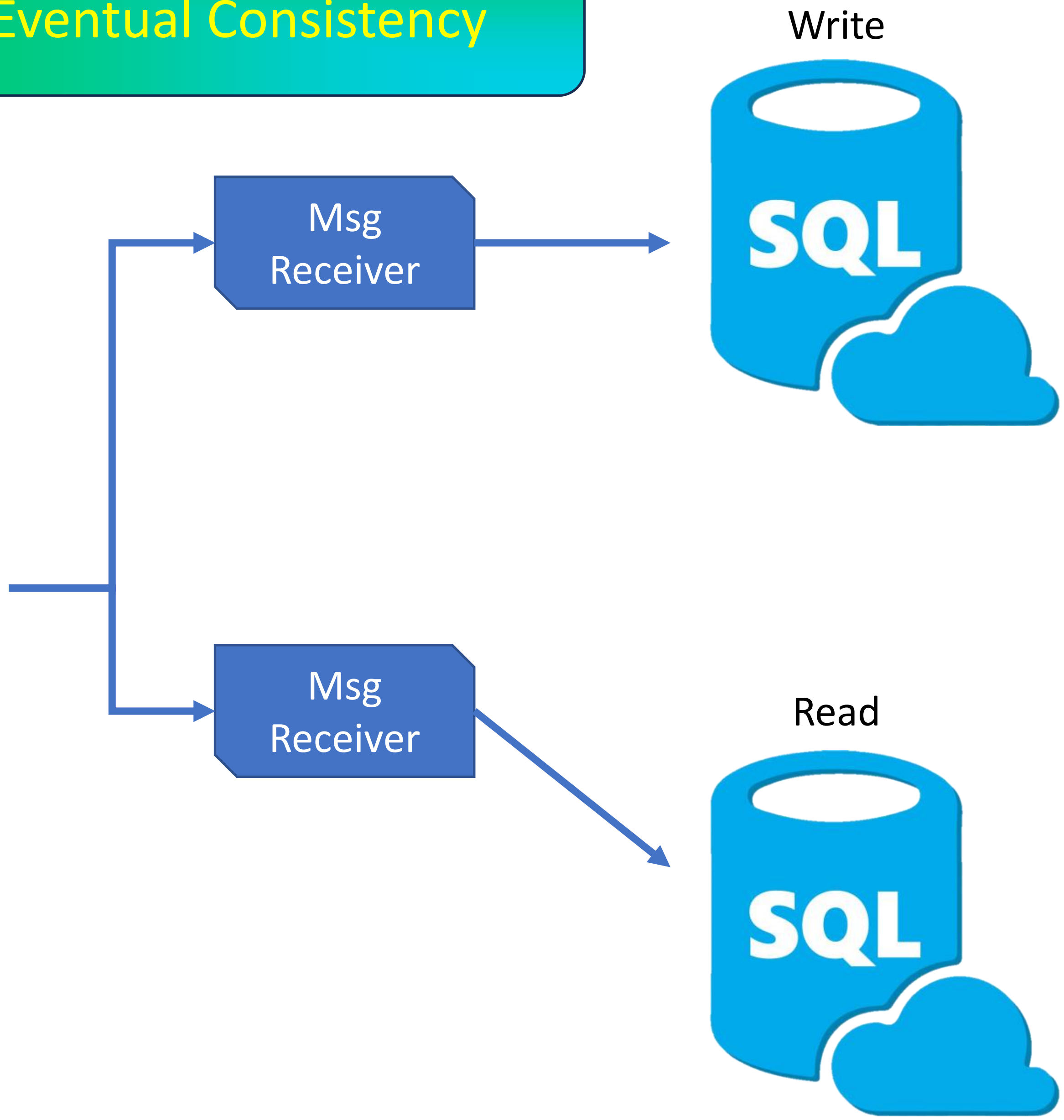
Msg Receiver

Read

1
2
3
4

1
2
3

Eventual Consistency



Write

Msg Receiver

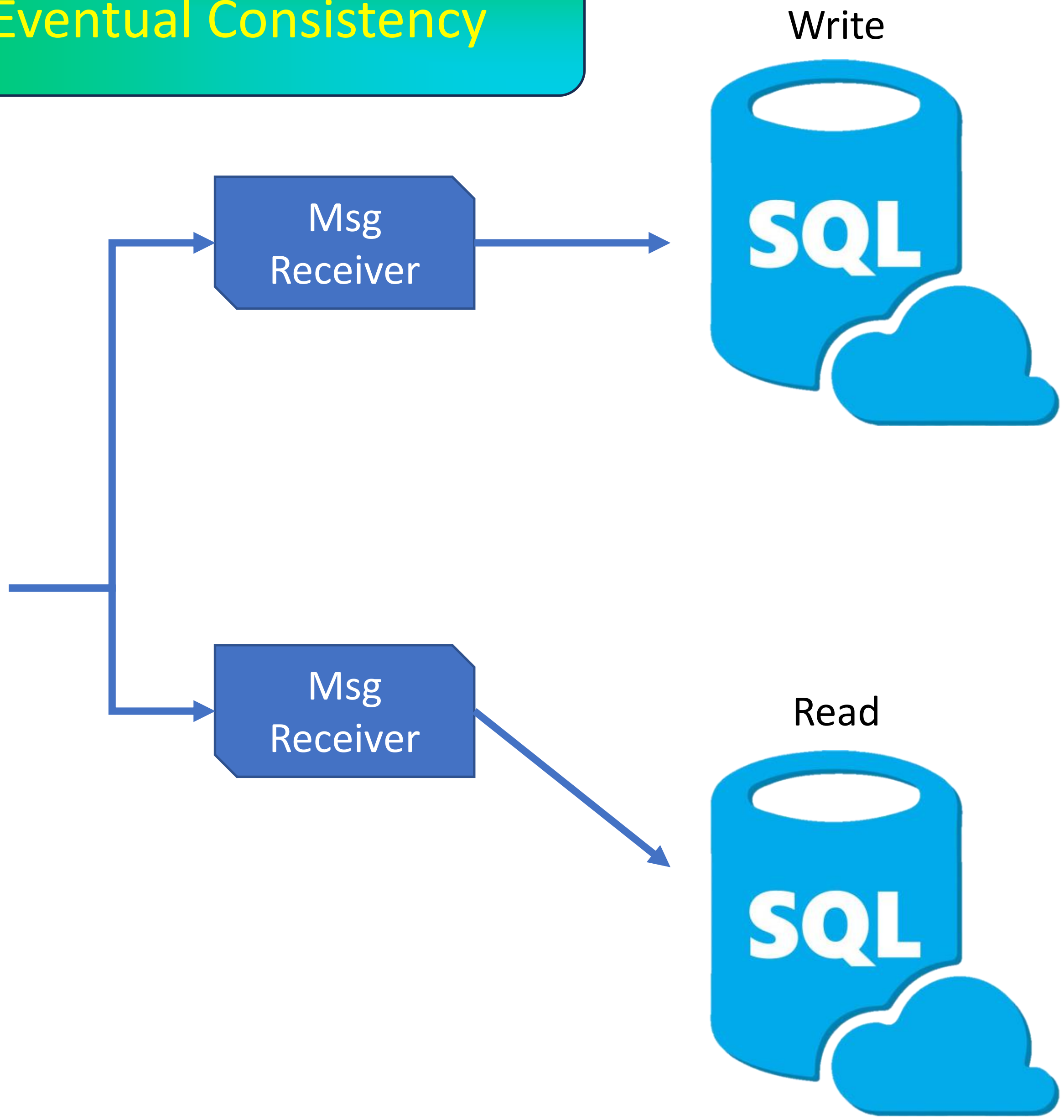
Msg Receiver

Read

1
2
3
4
5

1
2
3
4

Eventual Consistency

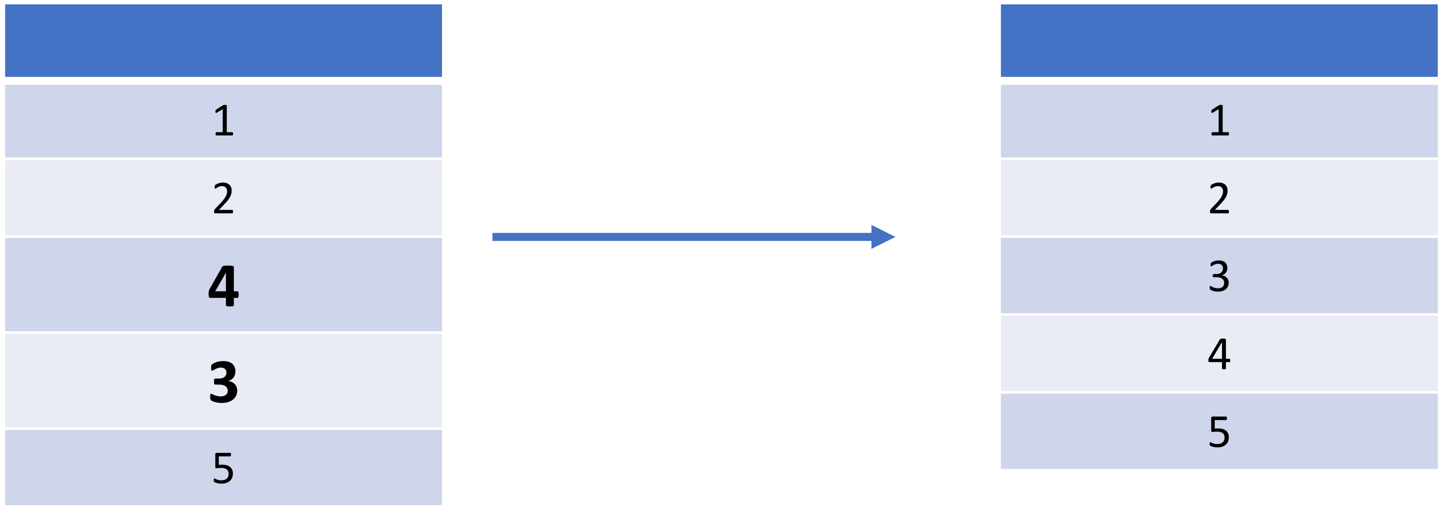


1
2
3
4
5

1
2
3
4
5

Select * From blah

Read



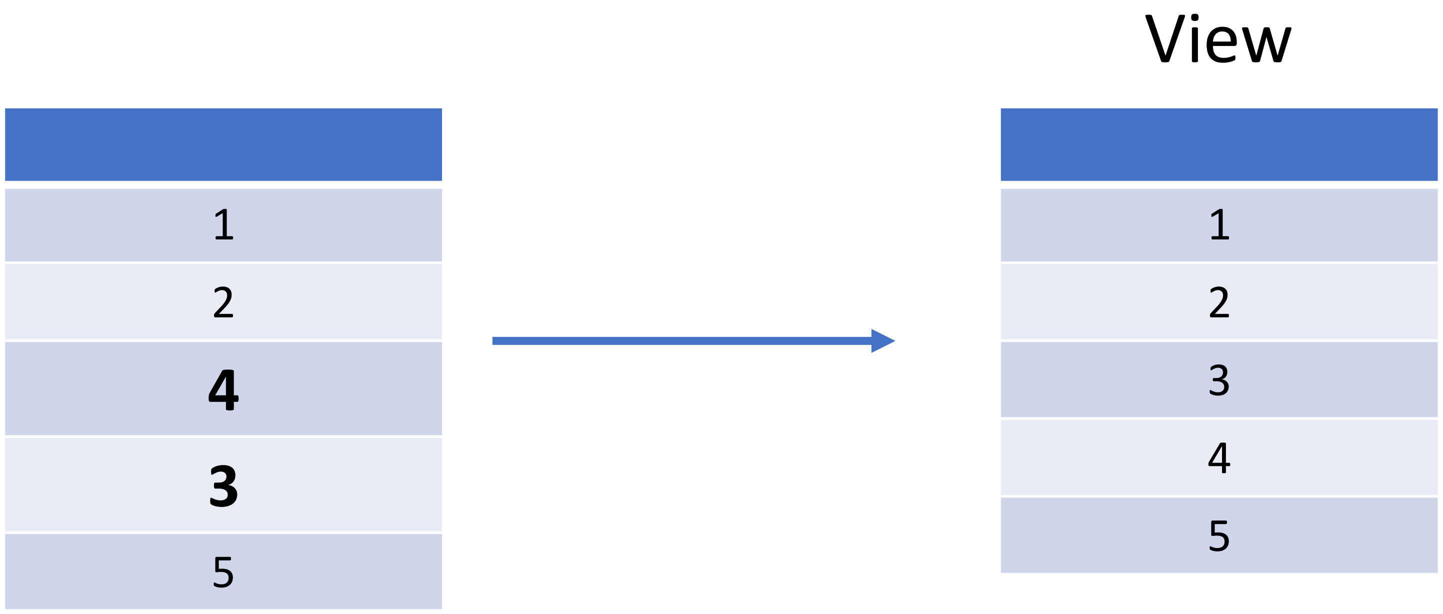
1
2
4
3
5

1
2
3
4
5

View

Select * From blah

Read

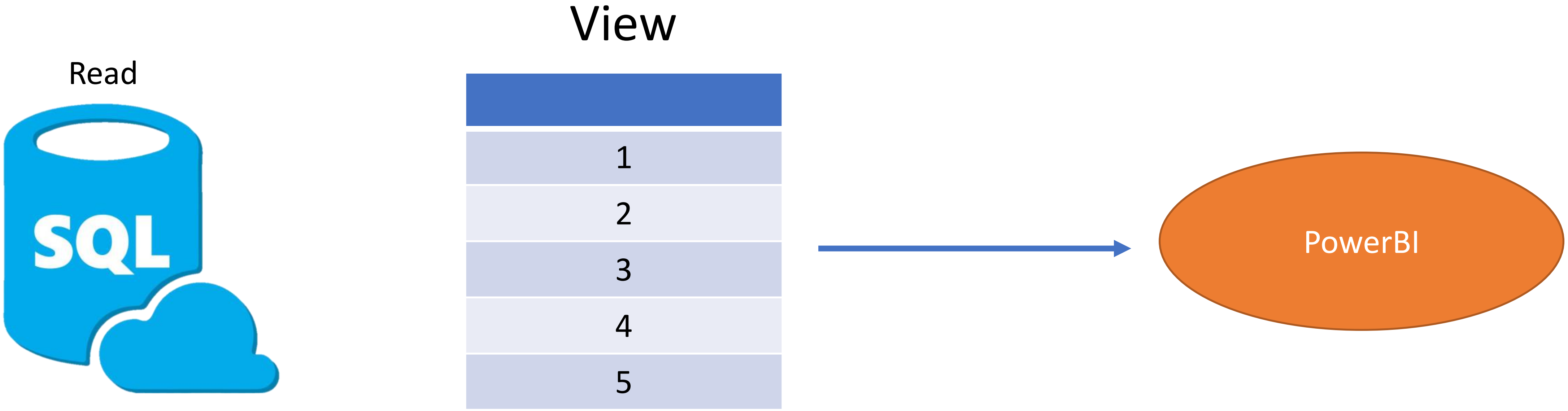


1
2
4
3
5

View

1
2
3
4
5

View



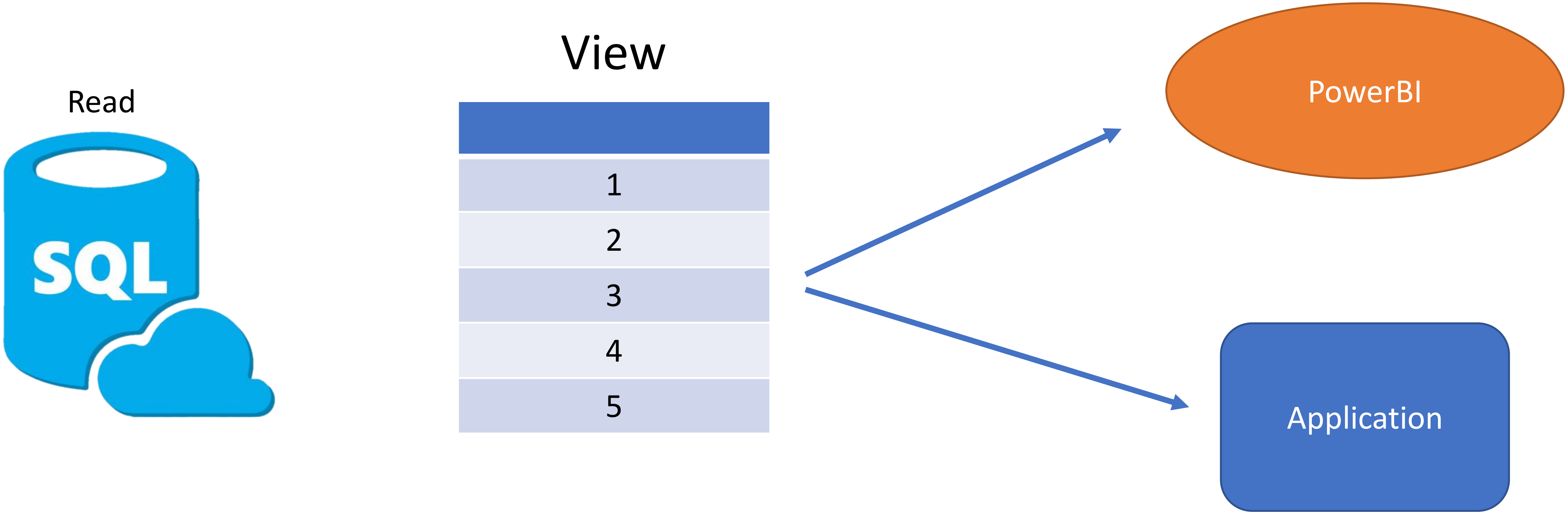
Read

View

1
2
3
4
5

PowerBI

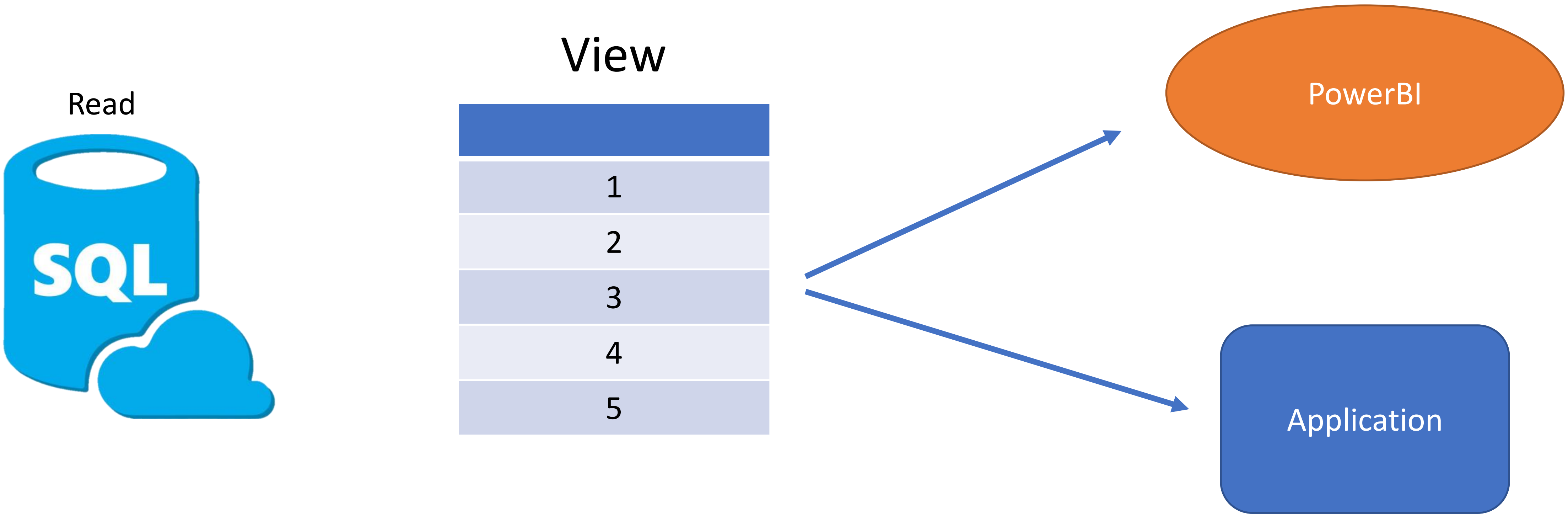
View



1
2
3
4
5

View

A *view* executes the query for every request.



Read

View

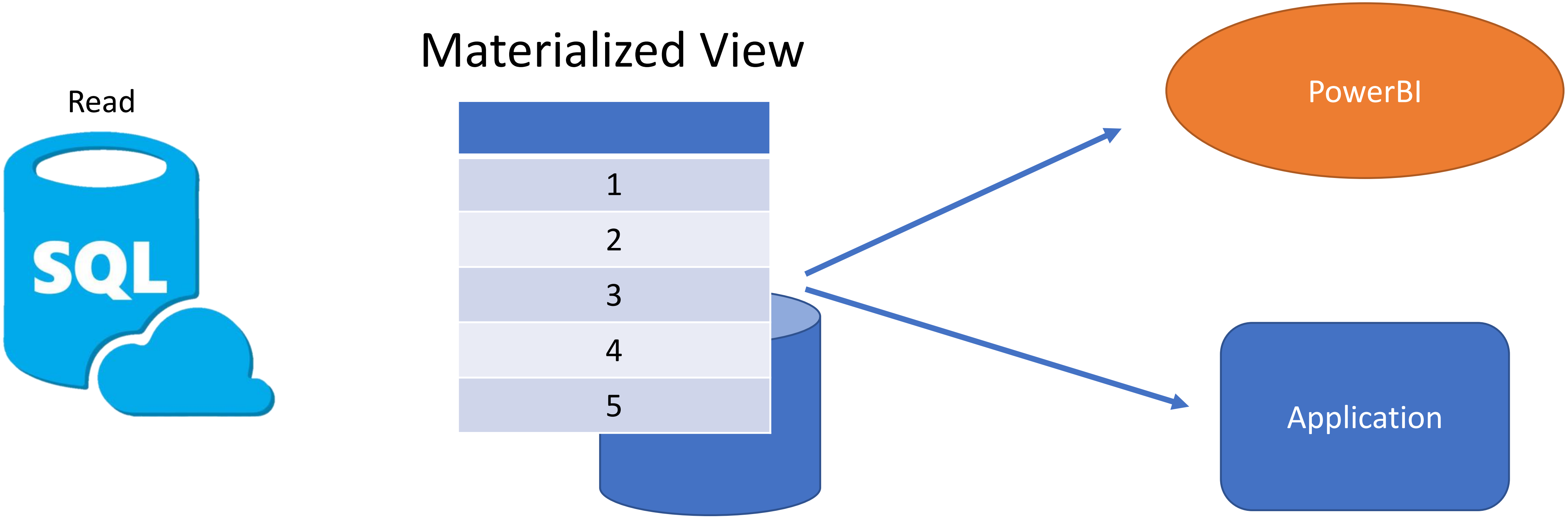
1
2
3
4
5

PowerBI

Application

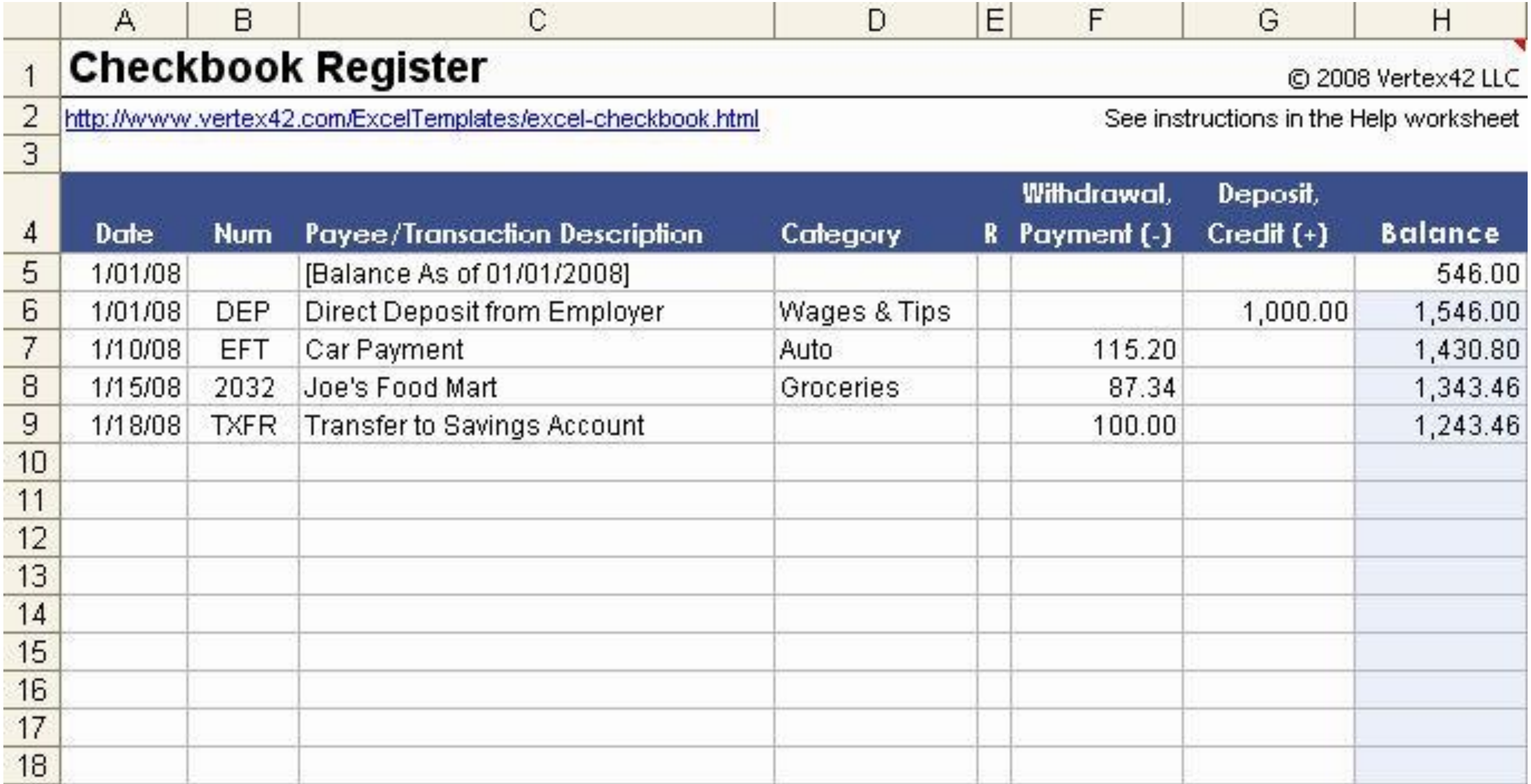
Materialized View

A *materialized view* is a form of a cache.



Event Sourcing

Event Sourcing



	A	B	C	D	E	F	G	H
1	**Checkbook Register**							© 2008 Vertex42 LLC
2	http://www.vertex42.com/ExcelTemplates/excel-checkbook.html							See instructions in the Help worksheet
3								
4	**Date**	**Num**	**Payee/Transaction Description**	**Category**	**R**	**Withdrawal, Payment (-)**	**Deposit, Credit (+)**	**Balance**
5	1/01/08		[Balance As of 01/01/2008]					546.00
6	1/01/08	DEP	Direct Deposit from Employer	Wages & Tips			1,000.00	1,546.00
7	1/10/08	EFT	Car Payment	Auto		115.20		1,430.80
8	1/15/08	2032	Joe's Food Mart	Groceries		87.34		1,343.46
9	1/18/08	TXFR	Transfer to Savings Account			100.00		1,243.46
10								
11								
12								
13								
14								
15								
16								
17								
18								

Event Sourcing

1/1/2023	5,000

Event Sourcing

1/1/2023	5,000
1/5/2023	100
1/5/2023	-50

Event Sourcing

1/1/2023	5,000
1/5/2023	100
1/5/2023	-50
1/15/2023	3,000

Event Sourcing

1/1/2023	5,000
1/5/2023	100
1/5/2023	-50
1/15/2023	3,000
1/16/2023	-2,400

Event Sourcing

1/1/2023	5,000
1/5/2023	100
1/5/2023	-50
1/15/2023	3,000
1/16/2023	-2,400
1/20/2023	-4,300

Event Sourcing

1/1/2023	5,000
1/5/2023	100
1/5/2023	-50
1/15/2023	3,000
1/16/2023	-2,400
1/20/2023	-4,300

Ending Balance?

Event Sourcing

1/1/2023	5,000
1/5/2023	100
1/5/2023	-50
1/15/2023	3,000
1/16/2023	-2,400
1/20/2023	-4,300

~~Ending Balance?~~ 1,350

Balance as of … ?

Getting a balance is relative to when you query the data.

Event Sourcing

```
{
  "Shopping Cart":{
    "Id":"SC-C123-202305301907-001",
    "CustomerId":"C123",
    "CustomerName":"Sean Whitesell",
    “Action”:”ItemAdded”,
    "Item":{
        "ItemId":"CoffeeXYZ-12345",
        "Description":"100 ct K-cup coffee",
        "Price":"24.99"
      }
  }
}
```

*Immutable Record

Event Sourcing

```
{
  "Shopping Cart":{
    "Id":"SC-C123-202305301907-001",
    "CustomerId":"C123",
    "CustomerName":"Sean Whitesell",
    “Action”:”ItemAdded”,
    "Item":{
        "ItemId":“CoffeeXYZ-StirStraw123",
        "Description":"100 ct coffee stirring straws",
        "Price":“5.49"
      }
  }
}
```

Event Sourcing

```
{
  "Shopping Cart":{
    "Id":"SC-C123-202305301907-001",
    "CustomerId":"C123",
    "CustomerName":"Sean Whitesell",
    "Items":[{
        "ItemId":"CoffeeXYZ-12345",
        "Description":"100 ct K-cup coffee",
        "Price":"24.99"
      },
      {
        "ItemId":“CoffeeXYZ-StirStraw123",
        "Description":"100 ct coffee stirring straws",
        "Price":“5.49"
      } ]
  }
}
```

View Cart

1. Add item Coffee
2. Add item Stir Straws

Event Sourcing

```
{
  "Shopping Cart":{
    "Id":"SC-C123-202305301907-001",
    "CustomerId":"C123",
    "CustomerName":"Sean Whitesell",
    “Action”:”ItemRemoved”,
    "Item":{
        "ItemId":“CoffeeXYZ-StirStraw123",
        "Description":"100 ct coffee stirring straws",
        "Price":“5.49"
      }
  }
}
```

Event Sourcing

```
{
  "Shopping Cart":{
    "Id":"SC-C123-202305301907-001",
    "CustomerId":"C123",
    "CustomerName":"Sean Whitesell",
    "Items":[{
            "ItemId":"CoffeeXYZ-12345",
            "Description":"100 ct K-cup coffee",
            "Price":"24.99"
        }]
  }
}
```

View Cart

1. Add item Coffee
2. Add item Stir Straws
3. Remove item Stir Straws

Event Sourcing

```
{
  "Shopping Cart":{
    "Id":"SC-C123-202305301907-001",
    "CustomerId":"C123",
    "CustomerName":"Sean Whitesell",
    “Action”:”ItemAdded”,
    "Item":{
        "ItemId":“CoffeeABC-StirStraw123",
        "Description":“200 ct coffee stirring straws",
        "Price":“6.49"
      }
  }
}
```

Event Sourcing

```
{
  "Shopping Cart":{
    "Id":"SC-C123-202305301907-001",
    "CustomerId":"C123",
    "CustomerName":"Sean Whitesell",
    "Items":[{
        "ItemId":"CoffeeXYZ-12345",
        "Description":"100 ct K-cup coffee",
        "Price":"24.99"
      },
      {
        "ItemId":“CoffeeABC-StirStraw123",
        "Description":“200 ct coffee stirring straws",
        "Price":“6.49"
      }]
  }
}
```

View Cart

1. Add item Coffee
2. Add item Stir Straws
3. Remove item Stir Straws
4. Add item Stir Straws

Event Sourcing

Forecasted and Actual Temperature Data

Time	Forecasted	Actual
6:00:00	76	77
7:00:00	76	75
8:00:00	65	62
9:00:00	60	56

Event Sourcing

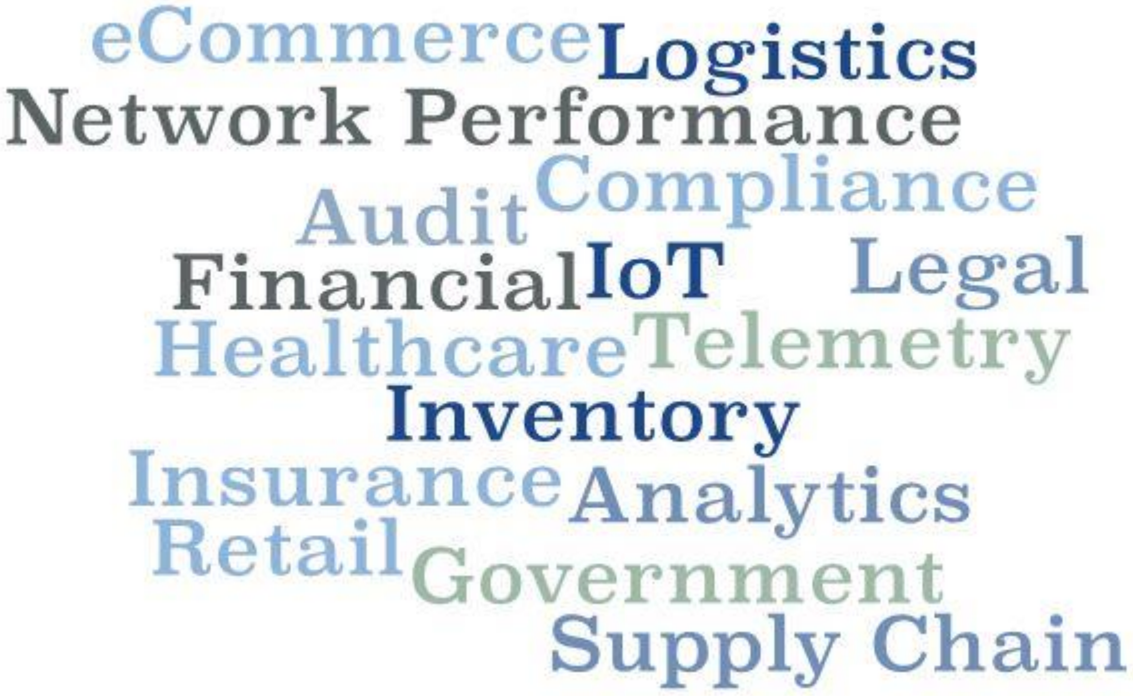
Analytics!

Forecasted and Actual Temperature Data

Time	Forecasted	Actual	Variation
6:00:00	76	77	1
7:00:00	76	75	-1
8:00:00	65	62	-3
9:00:00	60	56	-4

When to consider using Event Sourcing?

When to consider using Event Sourcing?



*not exhaustive list

Event Sourcing

Key point about Event Sourcing.

Event Sourcing

Key point about Event Sourcing.

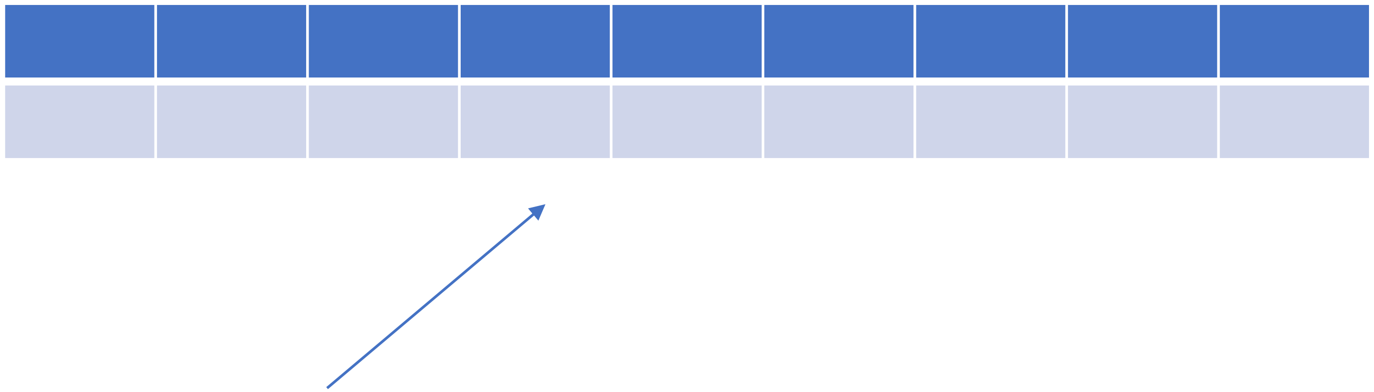
Use when you need to replay the change events as the “source of truth.”

Event Sourcing

Scenario

Event Sourcing

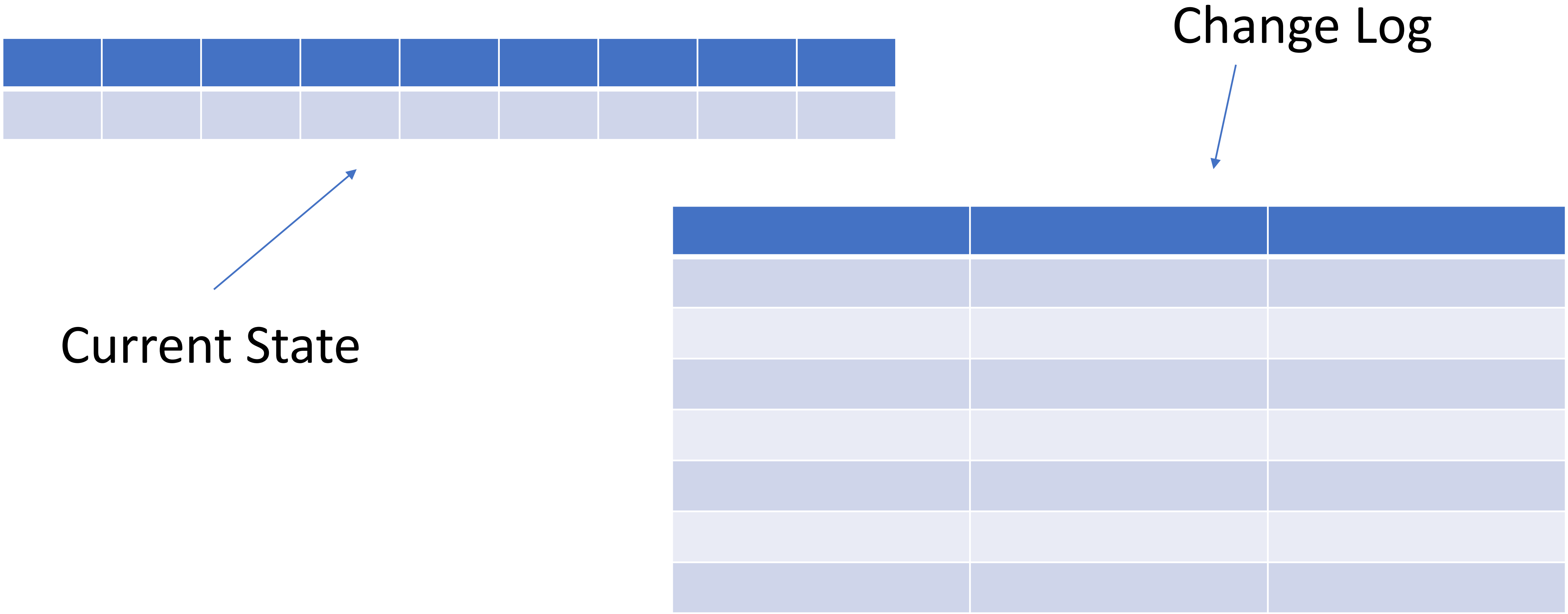
Scenario



Current State

Event Sourcing

Scenario

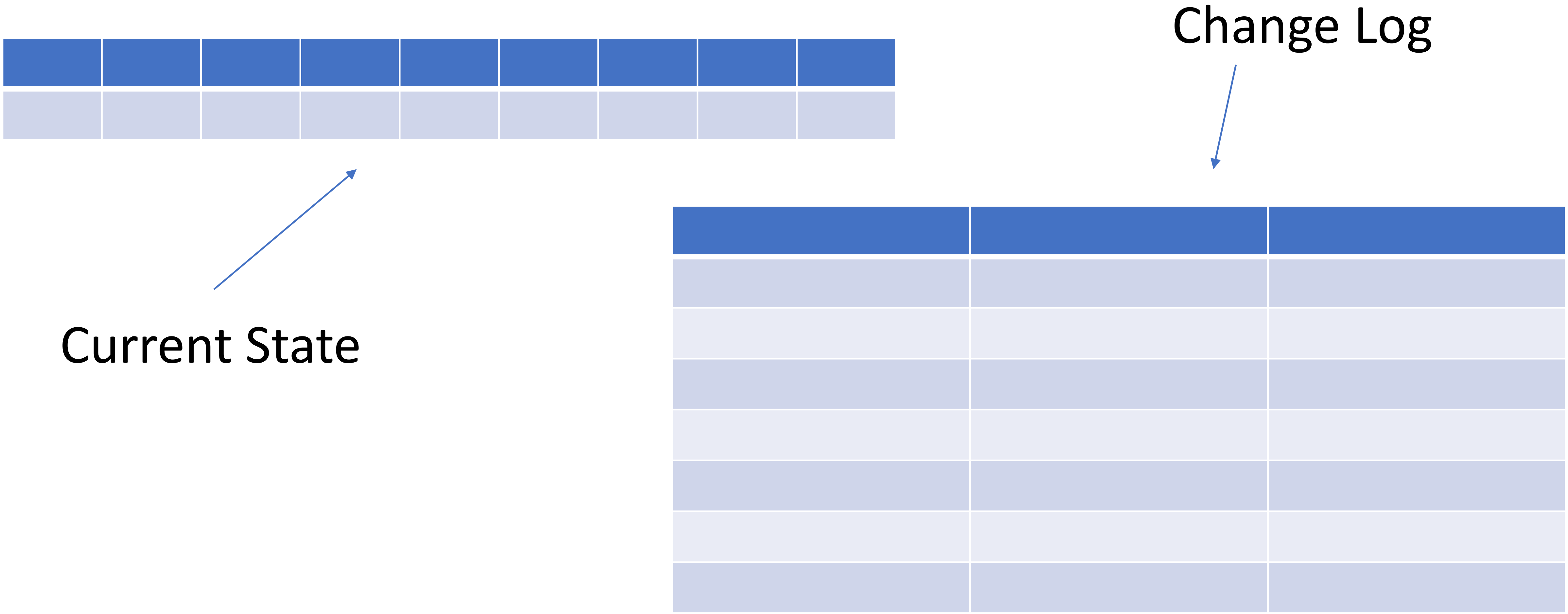


Change Log

Current State

Event Sourcing

Which one is the source of truth?

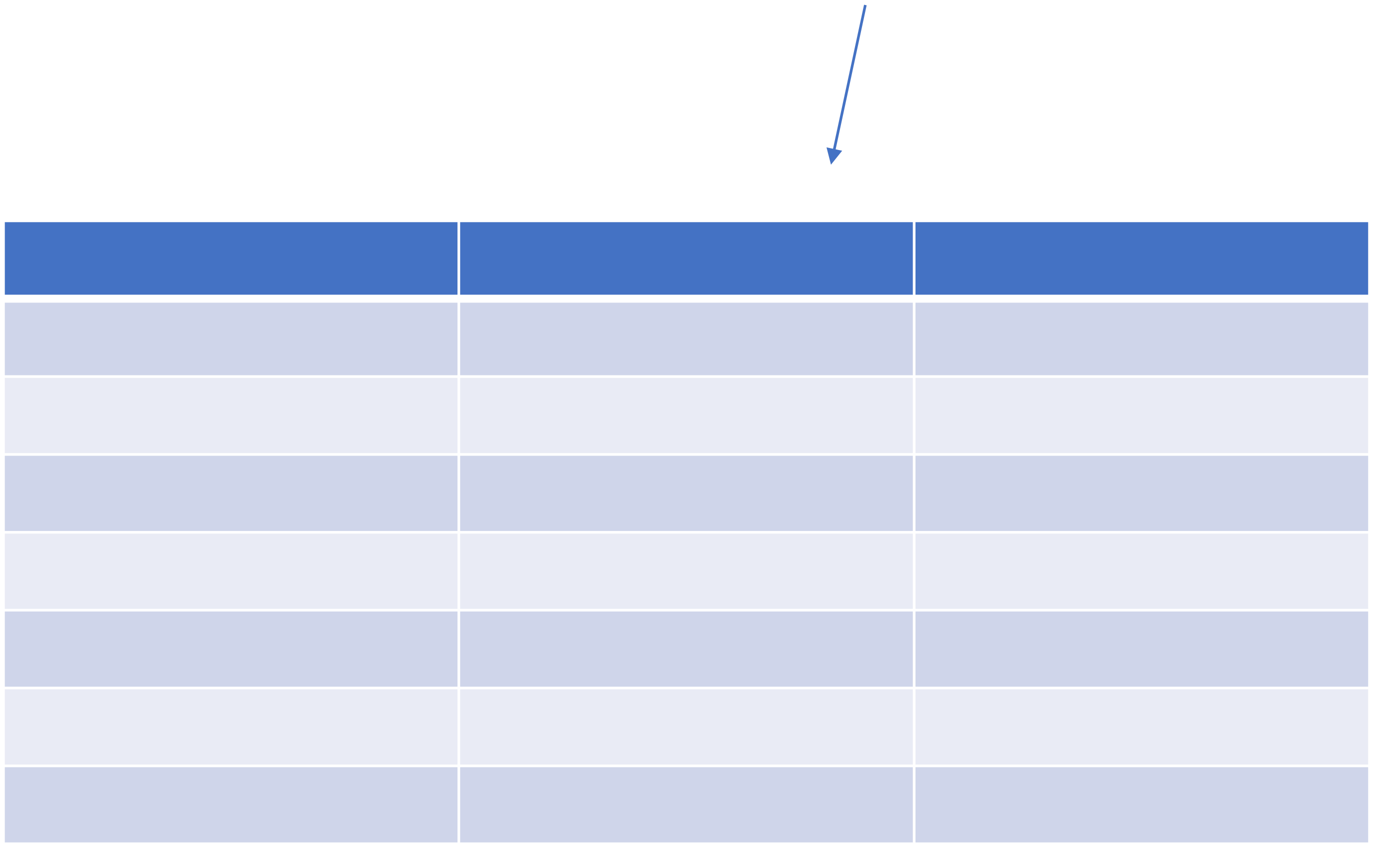


Change Log

Current State

Event Sourcing

Event Sourcing is having a change log *AS* the source of truth.

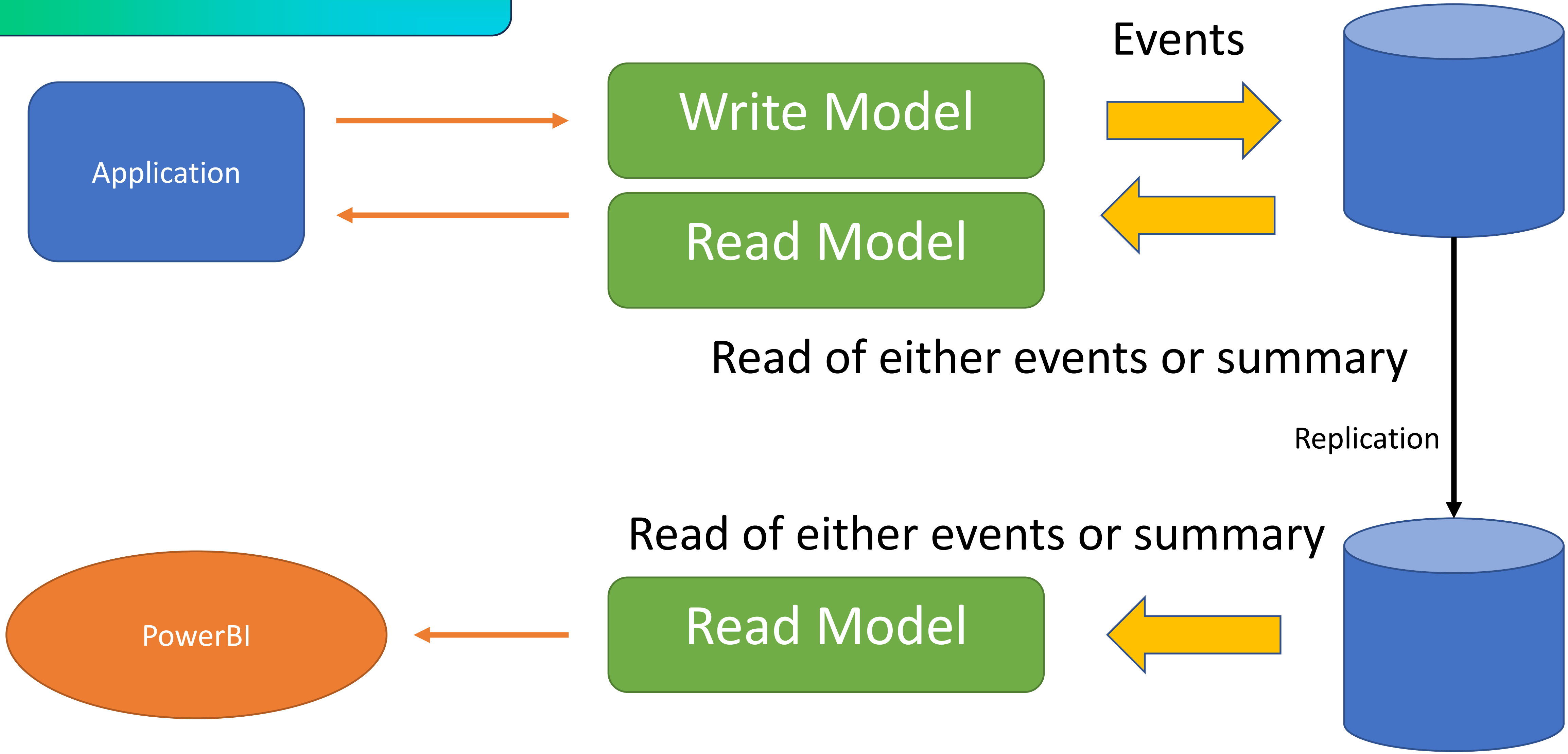


Change Log

Events of data changes

CQRS + Event Sourcing

C Q R S w/ Event Sourcing



Application

Write Model

Read Model

Events

Read of either events or summary

Replication

Read of either events or summary

PowerBI

Read Model

Data Security

Data Security

1. **At Rest vs In Transit**

Data that is stored on disk either spinning hard drive or solid-state is considered “*at rest*”.

Data in the network to or from a data store or between applications like microservices is considered “*in transit*”.

SSL connections is one way of encrypting data in transit between a source and destination.

Data Security

1. At Rest vs In Transit
2. **Key Management**

Data encrypted at rest uses an encryption key.

Most cloud providers do not allow you access to this key. Instead, they offer you a “customer managed key”. This CMK is a “*key encrypting key*.” It is used to encrypt the key that is used for encrypting the data at rest.

Data Security

1. At Rest vs In Transit
2. Key Management
3. **Access and Authorization**

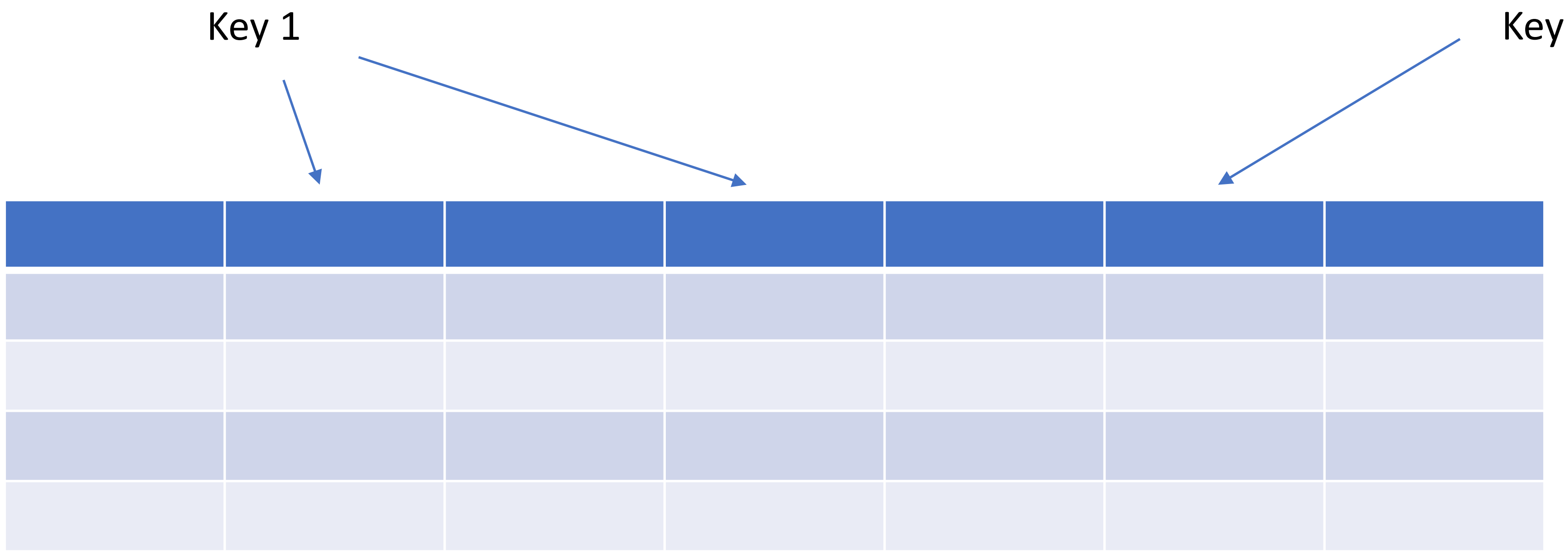
Some databases, like SQL Server, offer **Row-Level Security**.

This Row-Level Security allows for control of who has access to see any data at a row-by-row level.

Data Security

1. At Rest vs In Transit
2. Key Management
3. Access and Authorization
4. **Columnar Encryption**

Some databases, like SQL Server, offer **Transparent Data Encryption**. This uses a predefined key you set up for how the data in specified columns are encrypted.



Data Security

1. At Rest vs In Transit
2. Key Management
3. Access and Authorization
4. Columnar Encryption
5. **Compliance**

Data residency laws are becoming more prominent.

GDPR - General Data Protection Regulation
CCPA - California Consumer Privacy Act
PII – Personally Identifiable Information

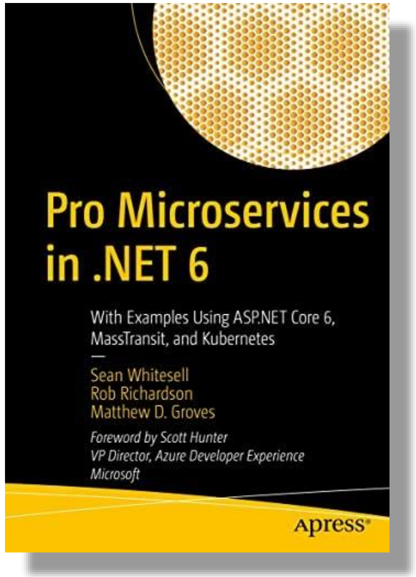
Questions?

github: seanw122/presentations

Sean Whitesell

President of Tulsa .NET User Group &&
Microsoft MVP &&
Sr. Cloud Architect @ ArchitectNow
Twitter: @codewithseanw

meetup.com/TulsaDevelopers-net



Pro Microservices in .NET 6

With Examples Using ASP.NET Core 6, MassTransit, and Kubernetes

Sean Whitesell
Rob Richardson
Matthew D. Groves

Foreword by Scott Hunter
VP Director, Azure Developer Experience
Microsoft

Apress®