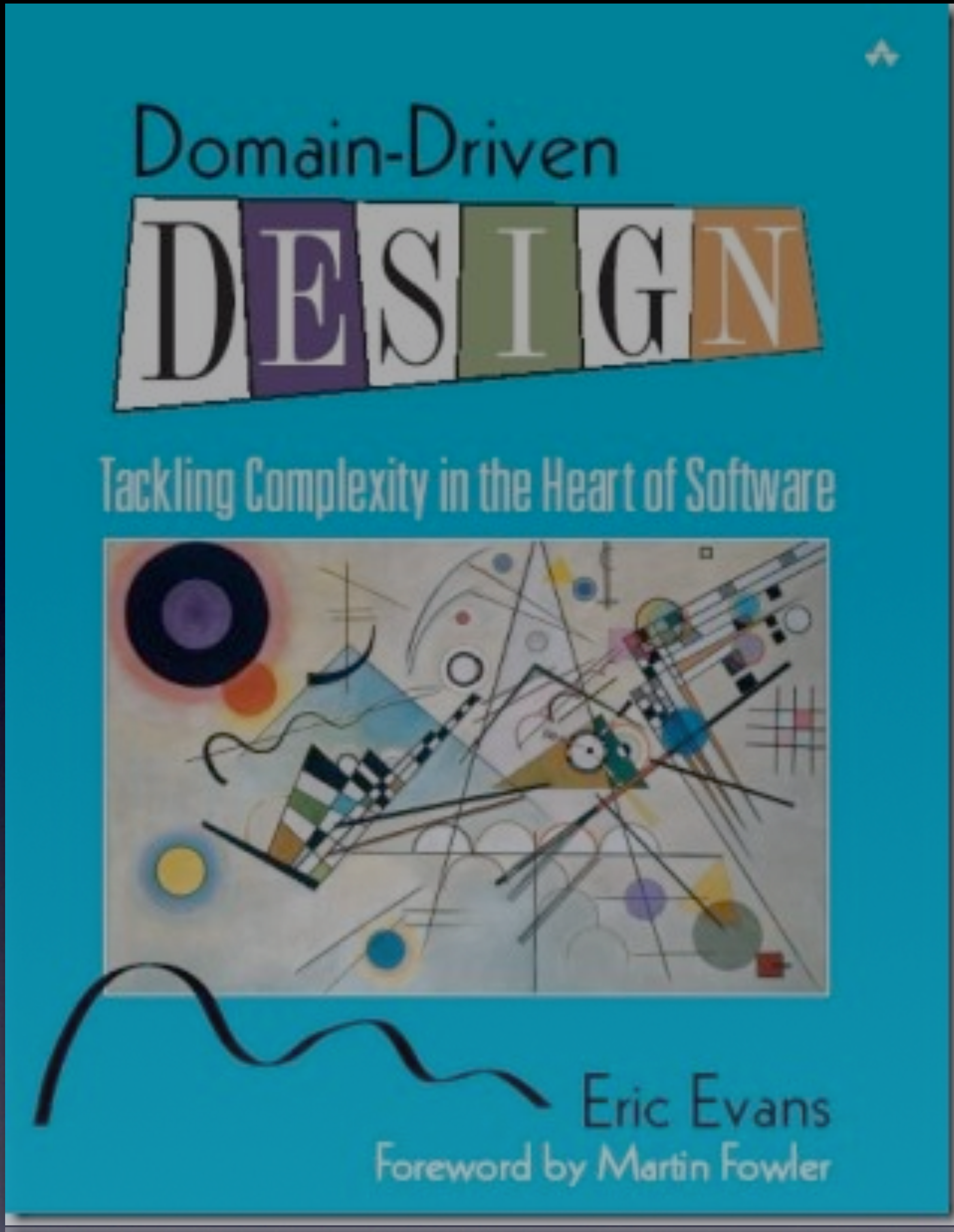
Domain-Driven Design and CQRS

Xebia ITR on
Command Query Responsibility Separation

Sjors Grijpink and Erik Rozendaal

DDD

- Eric Evans, Domain-Driven Design, 2004
- Key concepts
  - Ubiquitous Language
  - Value Object, Entity
  - Aggregate


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

Ubiquitous Language

- Language shared between domain experts and developers
- No need for error-prone translation
- Maps directly to domain implementation
- Implementation should be free of “technical” terms

Value Object

- No conceptual identity
- Describe characteristic of a thing
- Usually immutable
- Examples: Address, Money, ...

Entity

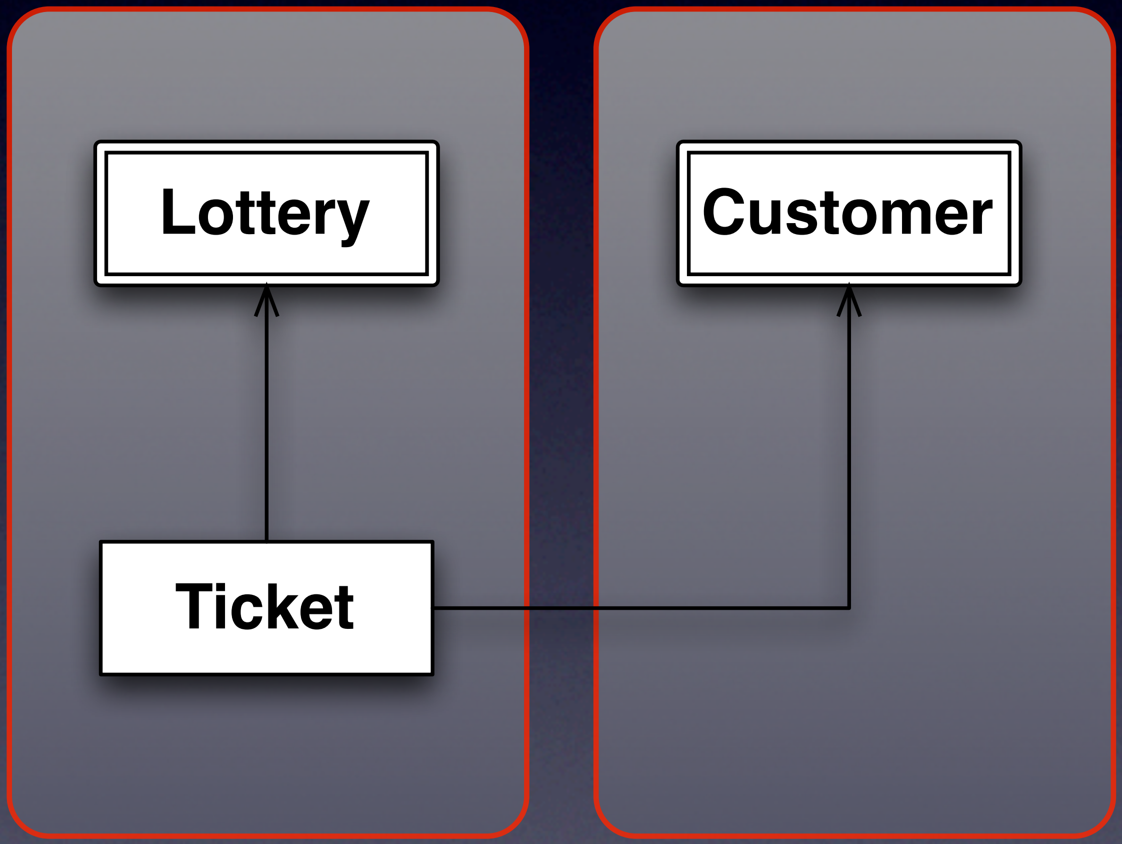
- Something with a unique identity
- Identity does not change when any of its attributes change
- Examples: Customer, Order, ...

Aggregate

- Group of Entities & Value Objects
- One entity within the aggregate is the aggregate root
- All access to the objects inside go through the root entity
- Aggregates are consistency boundaries

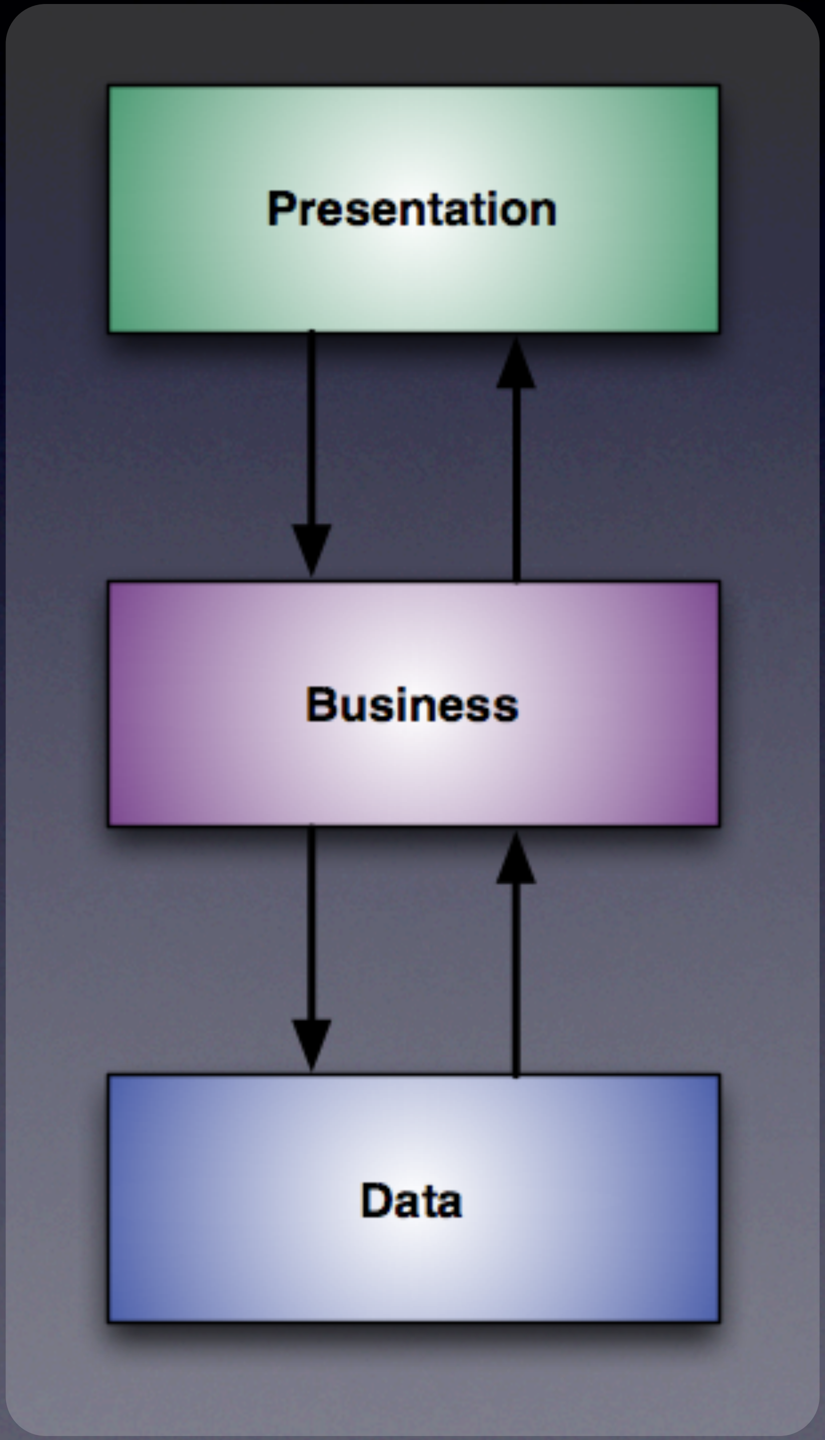
Lottery

- Two aggregate roots
  - Lottery
  - Customer

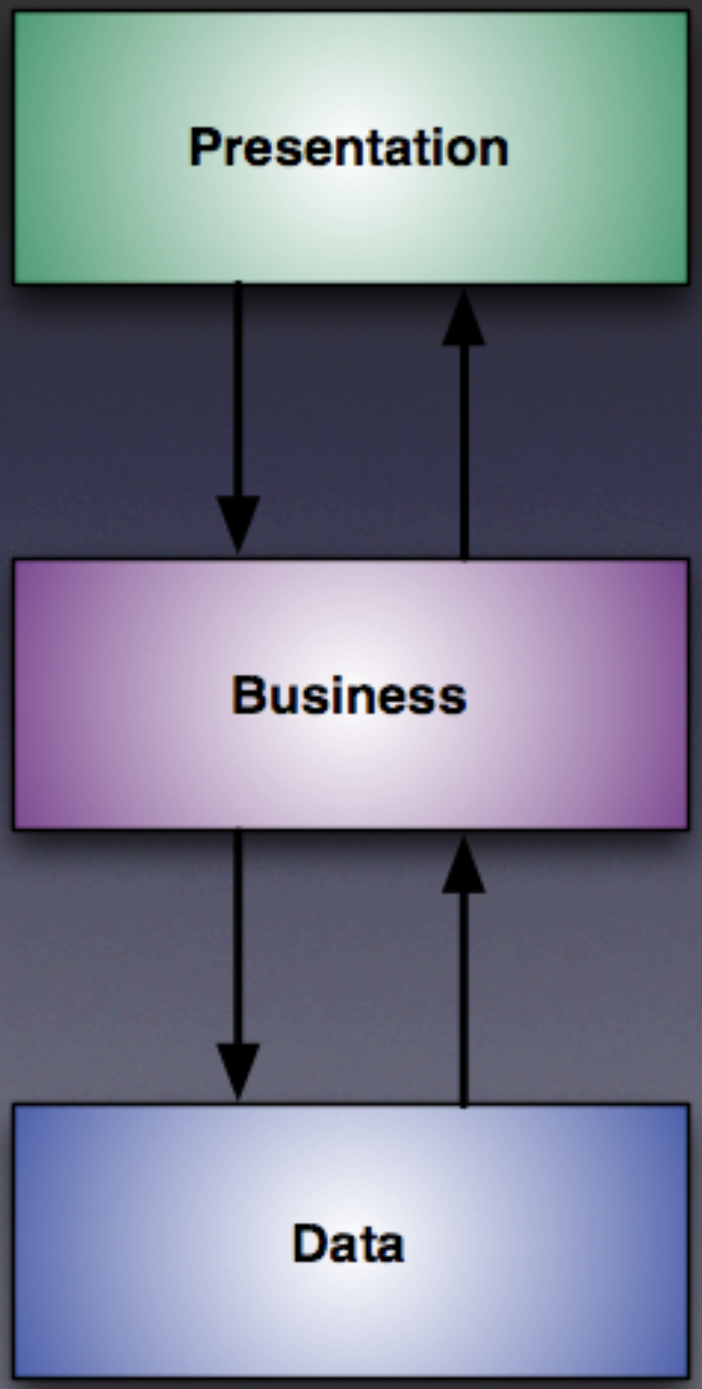


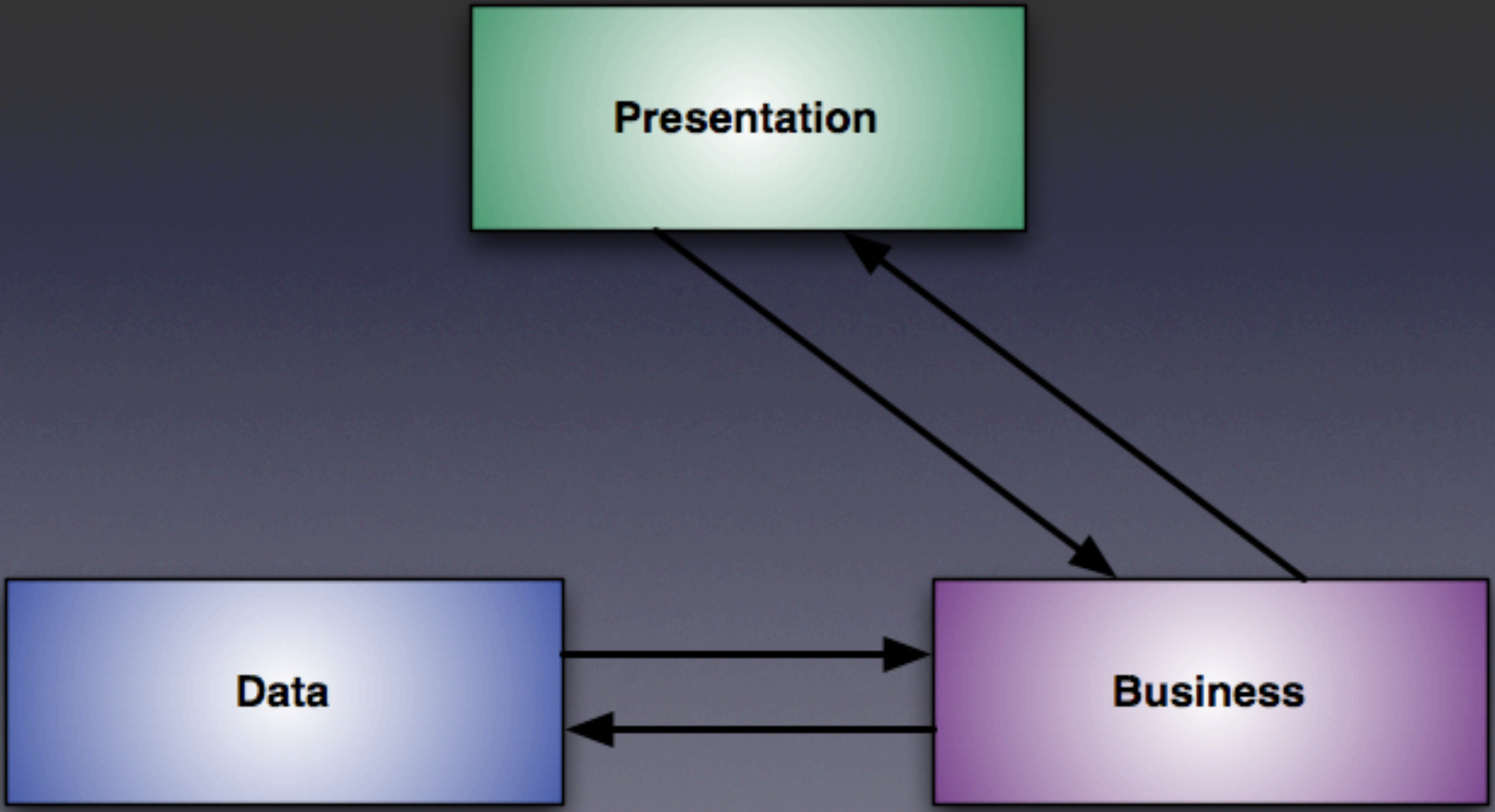
3 Tier

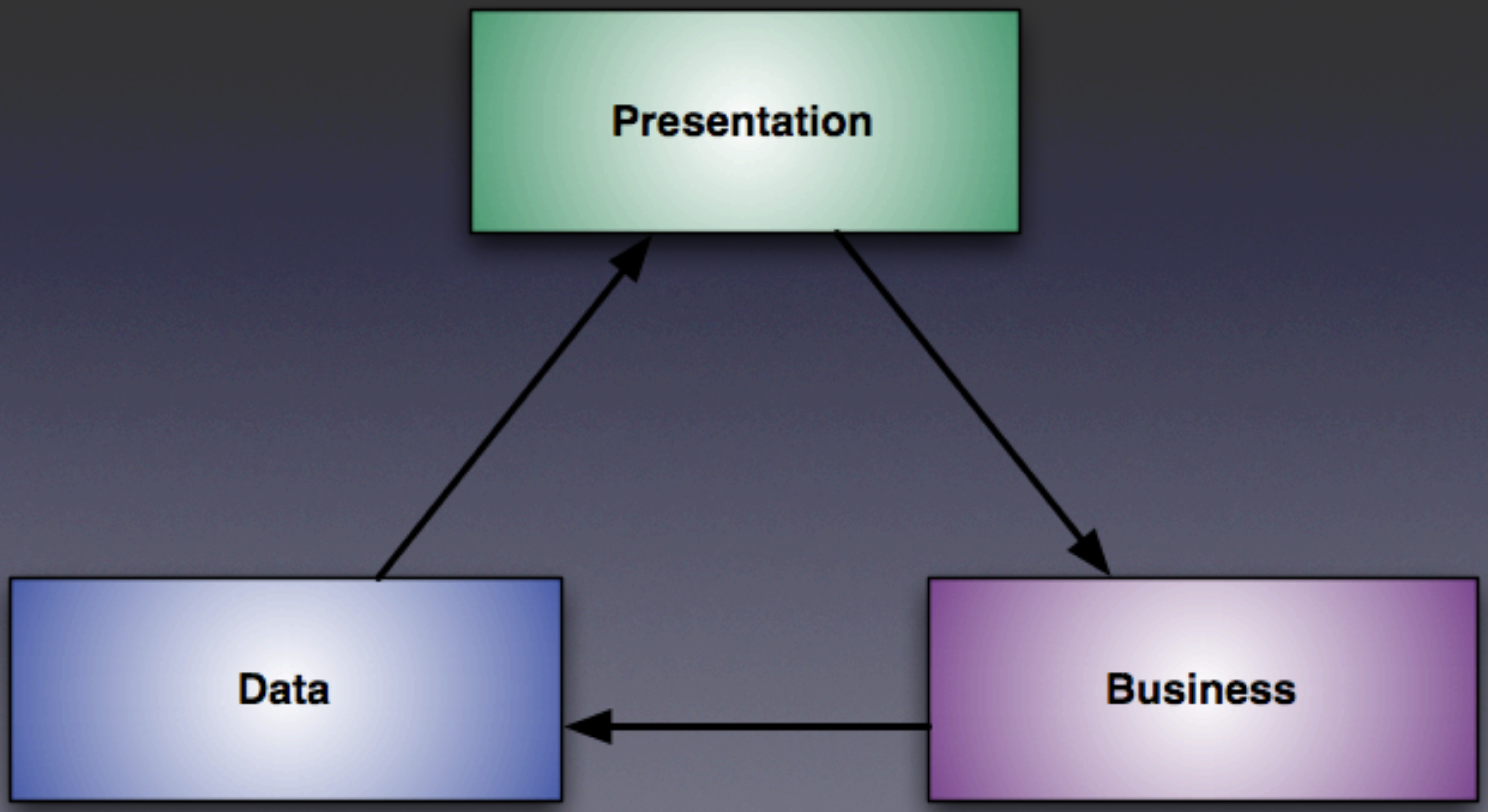
- Presentation layer is *denormalized*
- Domain is *behavioral*
- Transactional database is *normalized*

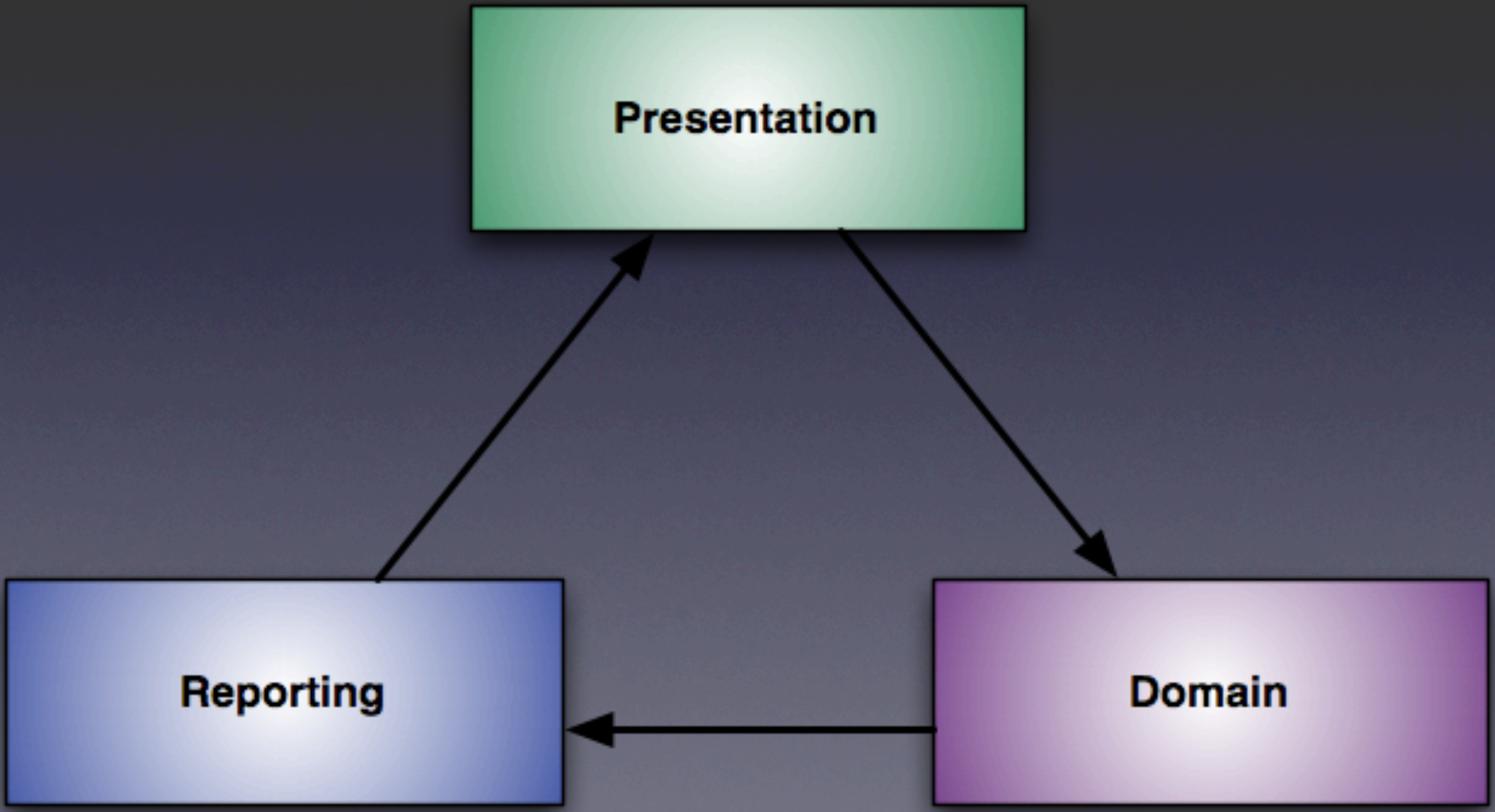


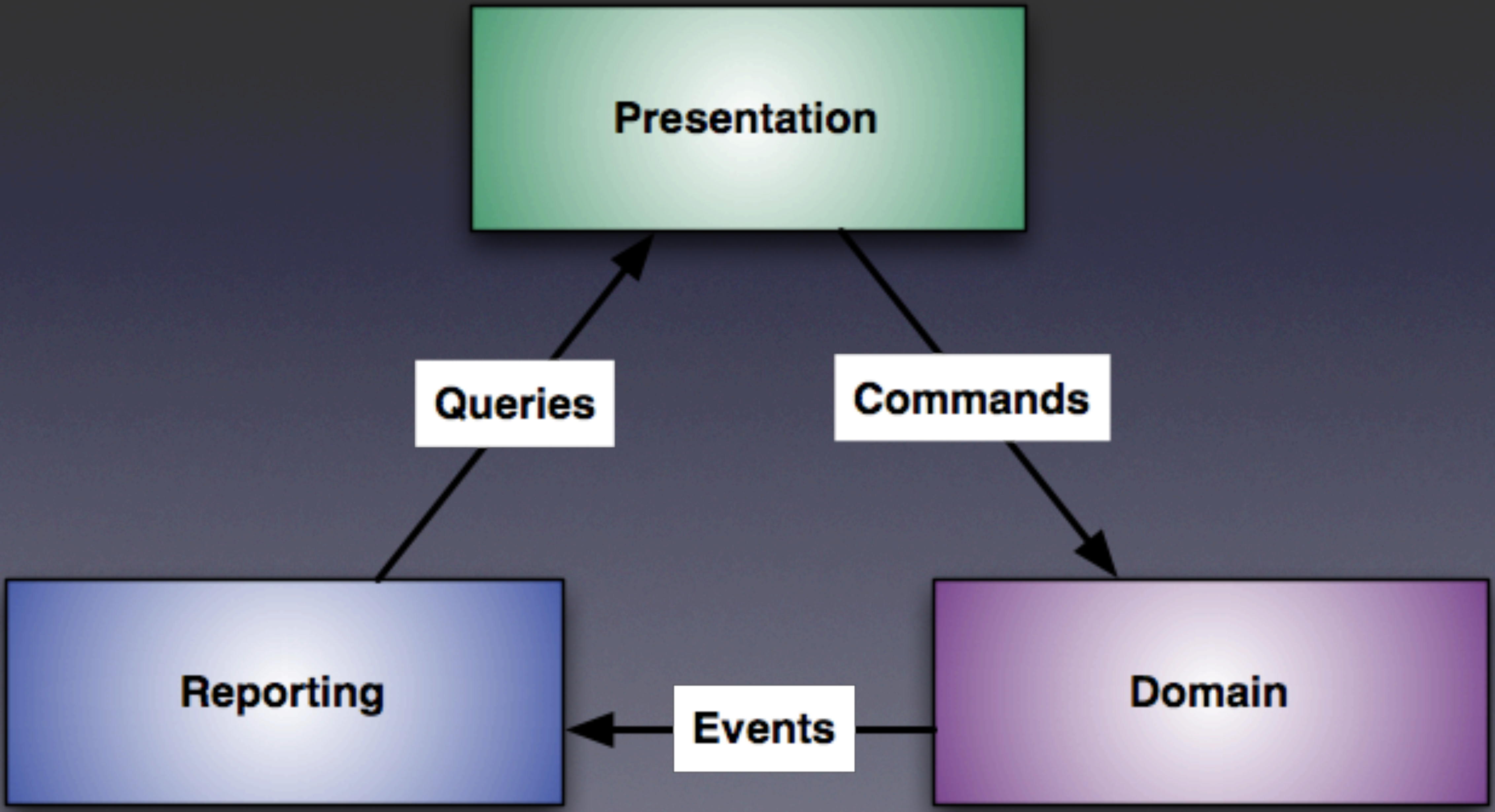
“A single model cannot be appropriate for reporting, searching, and transactional behaviors.” - Greg Young

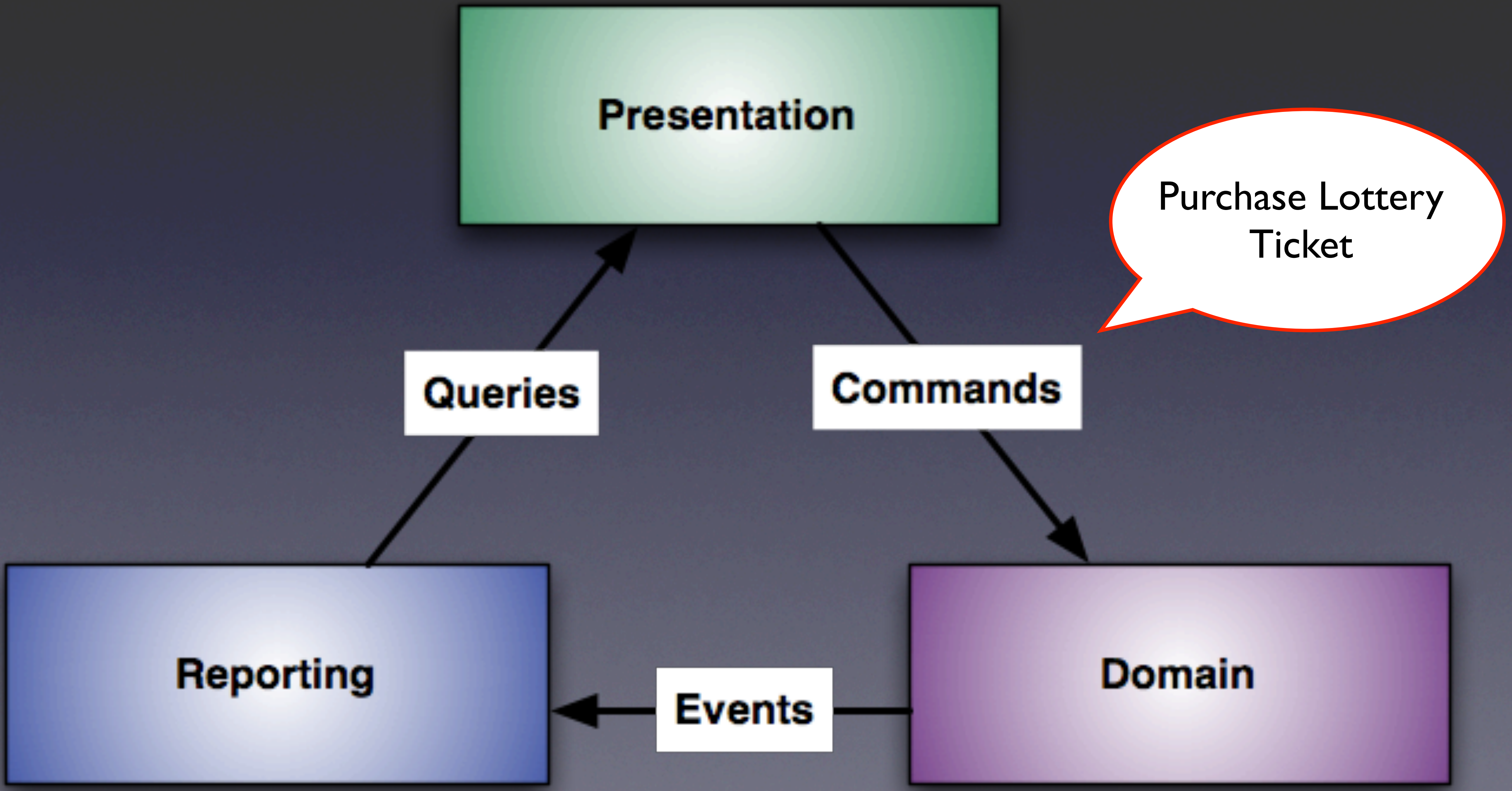

Presentation
Business
Data

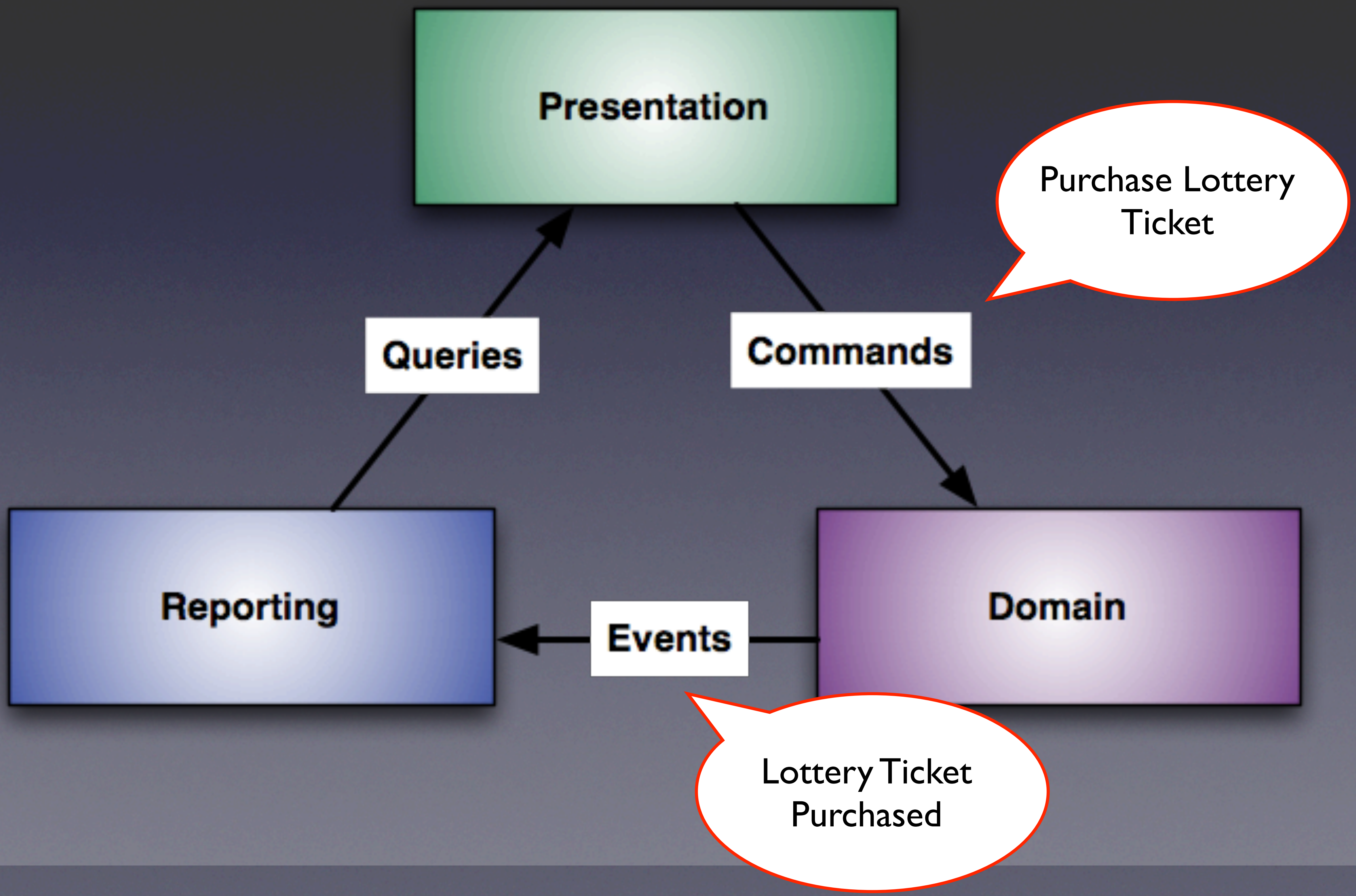

Presentation
Data
Business

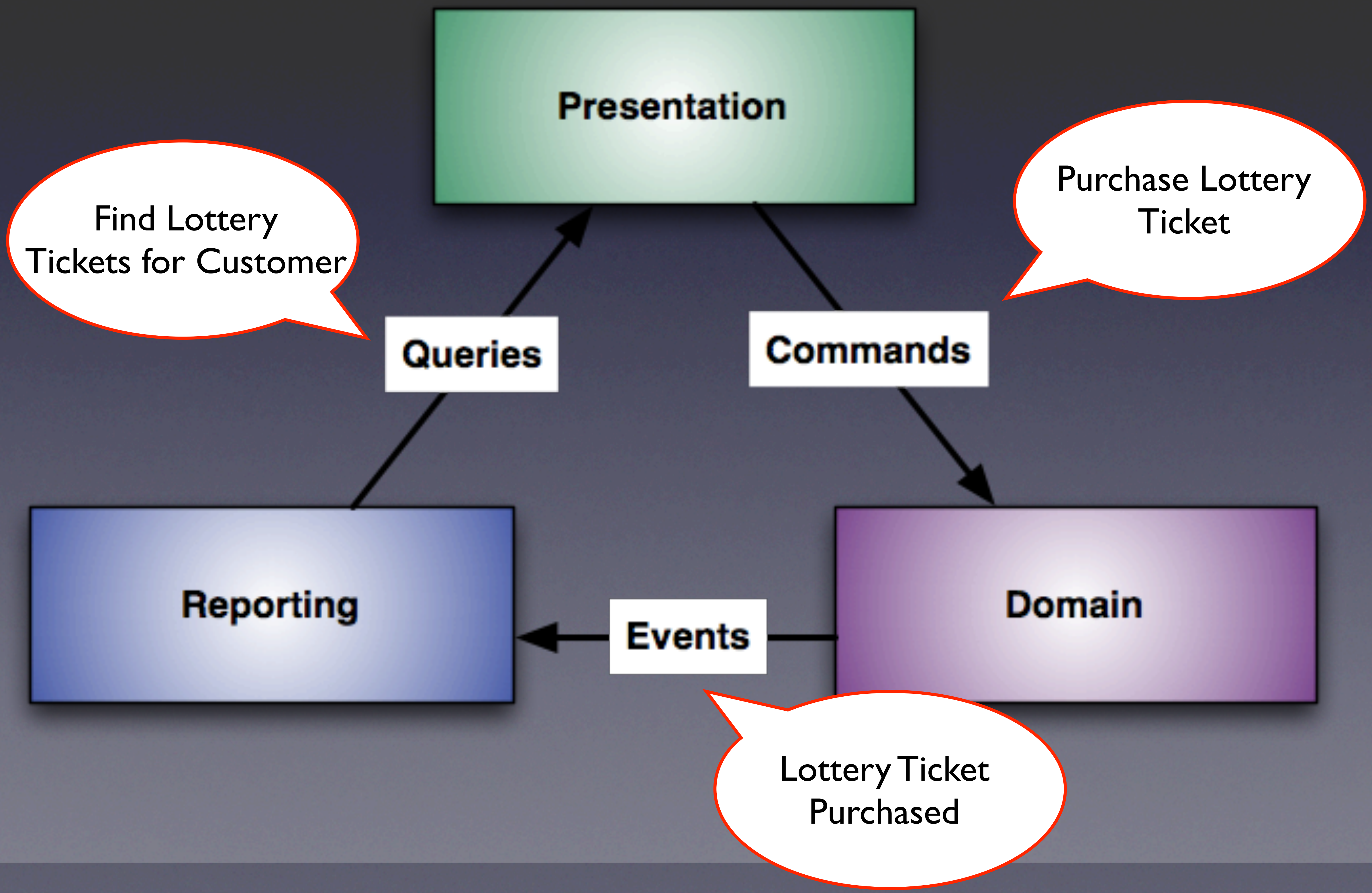

Presentation
Data
Business


Presentation
Reporting
Domain


Presentation
Queries
Commands
Reporting
Events
Domain


Presentation
Purchase Lottery Ticket
Queries
Commands
Reporting
Events
Domain


Presentation
Queries
Commands
Purchase Lottery Ticket
Reporting
Events
Domain
Lottery Ticket Purchased

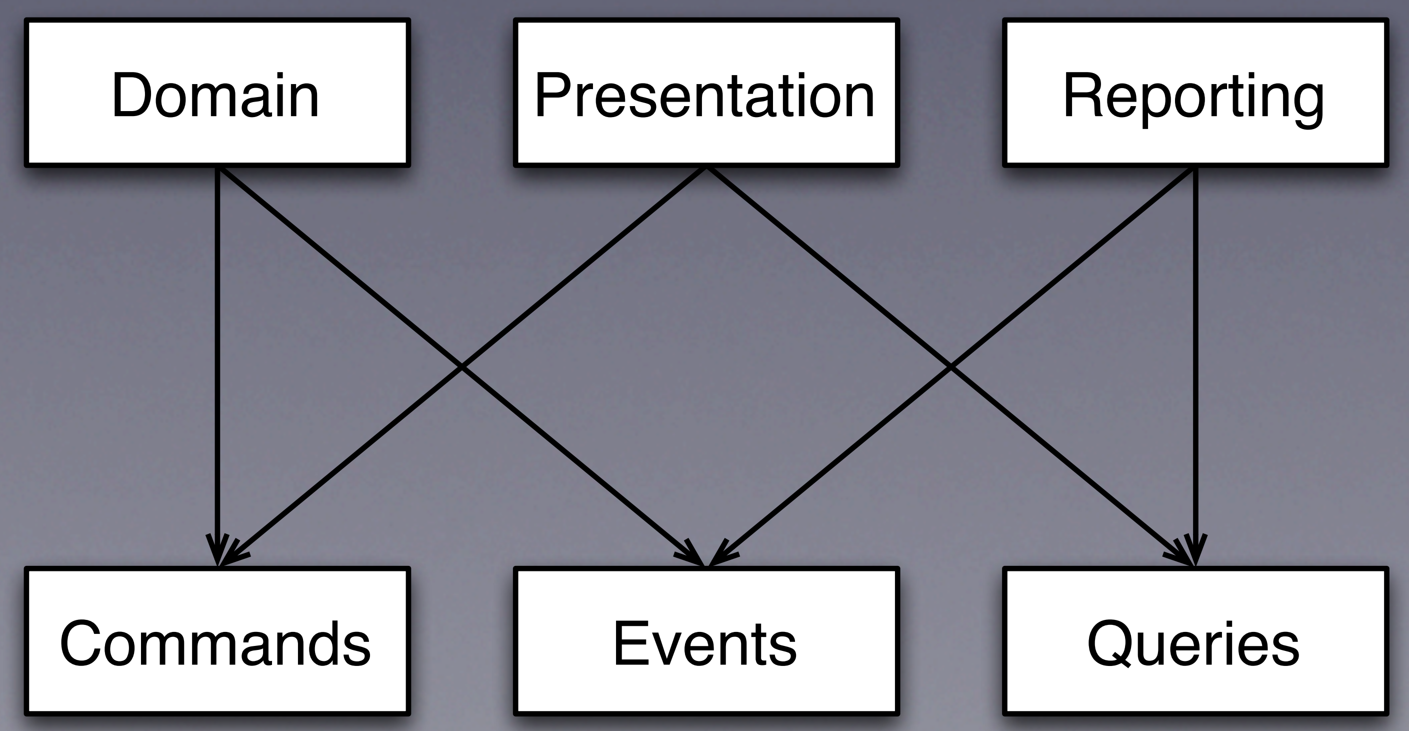

Presentation
Find Lottery Tickets for Customer
Queries
Commands
Purchase Lottery Ticket
Reporting
Events
Domain
Lottery Ticket Purchased

All state changes are represented by *Domain Events*

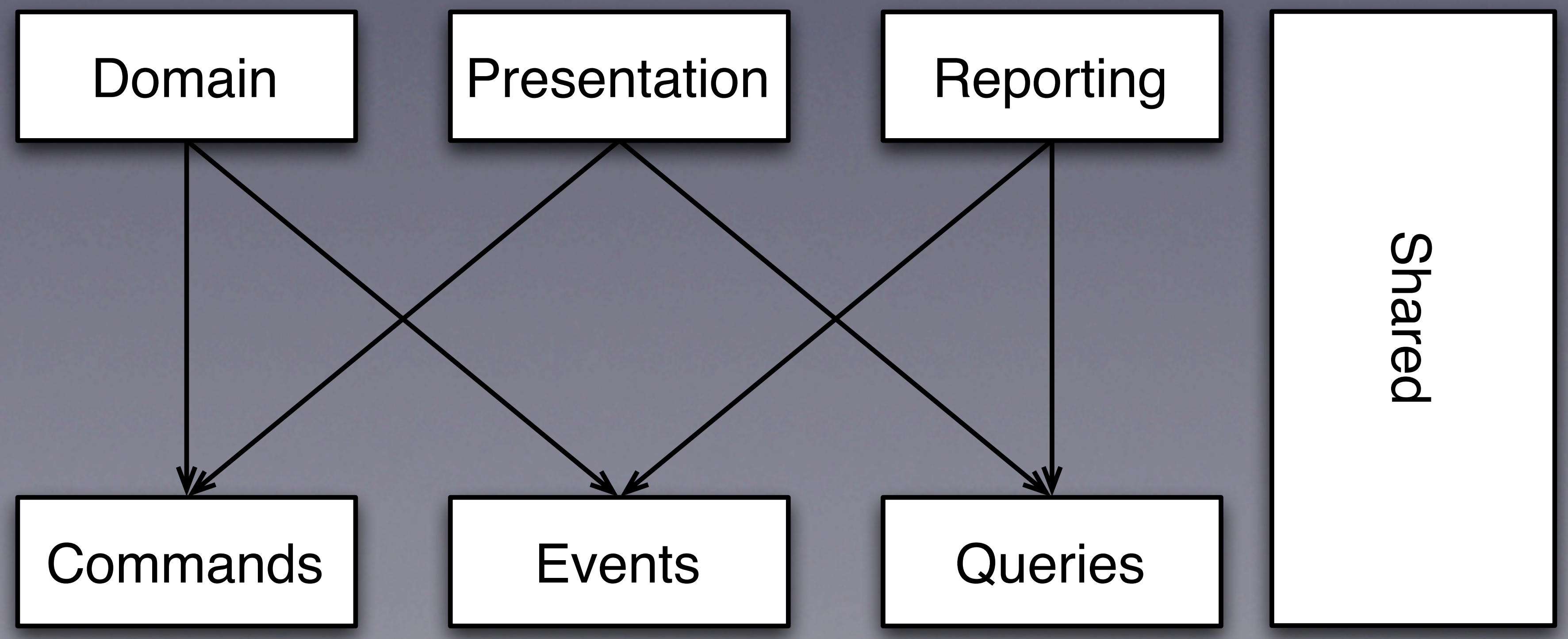
Domain

Reporting

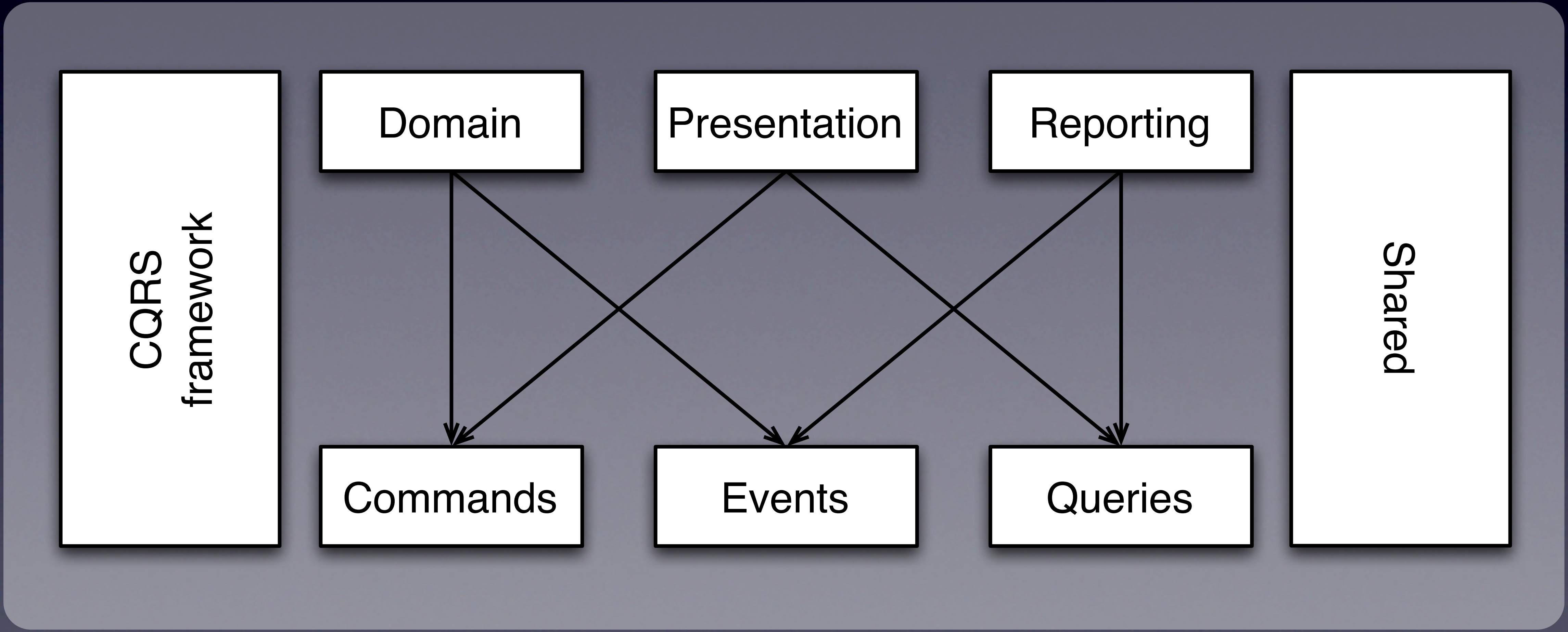
Packages



Packages


Domain
Presentation
Reporting
Shared
Commands
Events
Queries

Packages


CQRS framework
Domain
Presentation
Reporting
Shared
Commands
Events
Queries

Exercise 1

- Implement "Create Customer"
- Test that customers are listed on the screen
- Test that customers cannot be created with an initial payment of less than 10.00

Exercise 2

- Implement "Purchase Lottery Ticket"
- Ticket should be listed
- Customer's balance should be updated
- Don't forget to check the customer's account balance

Benefits

Benefits

- Fully encapsulated domain that only exposes behavior

Benefits

- Fully encapsulated domain that only exposes behavior
- Queries do not use the domain model

Benefits

- Fully encapsulated domain that only exposes behavior
- Queries do not use the domain model
- No object-relational impedance mismatch

Benefits

- Fully encapsulated domain that only exposes behavior
- Queries do not use the domain model
- No object-relational impedance mismatch
- Bullet-proof auditing and historical tracing

Benefits

- Fully encapsulated domain that only exposes behavior
- Queries do not use the domain model
- No object-relational impedance mismatch
- Bullet-proof auditing and historical tracing
- Easy integration with external systems

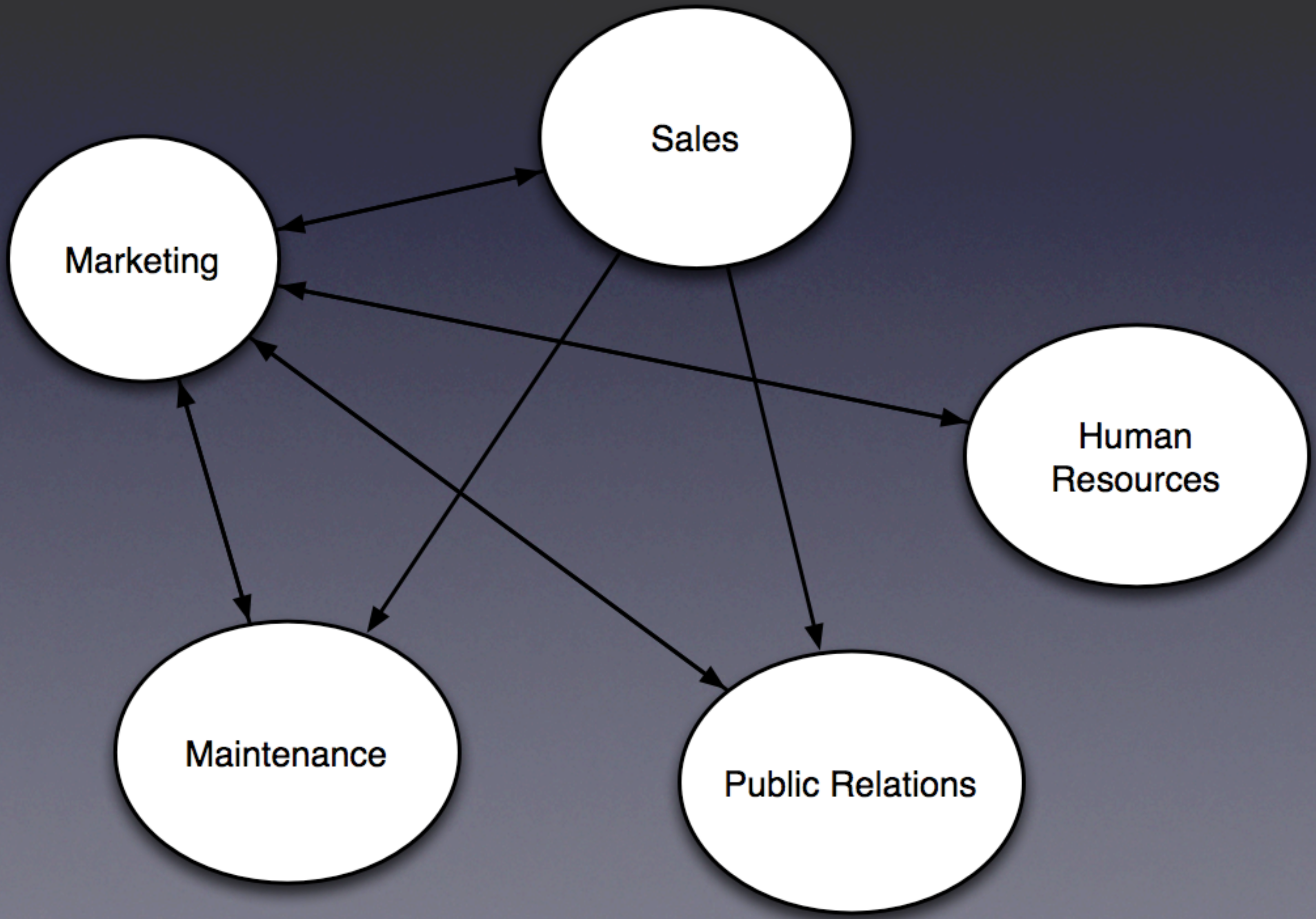
Benefits

- Fully encapsulated domain that only exposes behavior
- Queries do not use the domain model
- No object-relational impedance mismatch
- Bullet-proof auditing and historical tracing
- Easy integration with external systems
- Performance and scalability

Benefits

- Fully encapsulated domain that only exposes behavior
- Queries do not use the domain model
- No object-relational impedance mismatch
- Bullet-proof auditing and historical tracing
- Easy integration with external systems
- Performance and scalability
- Testability

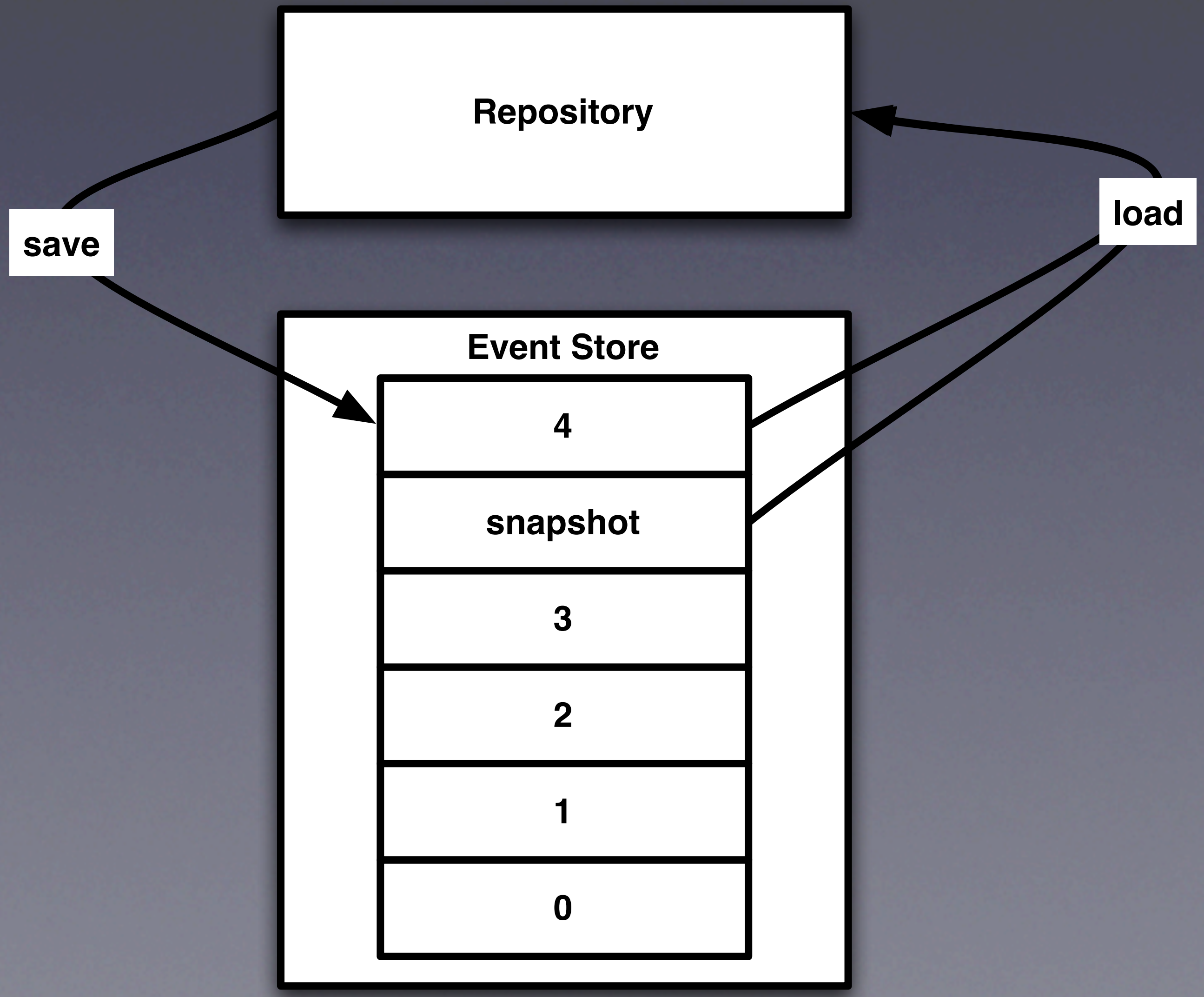
Bounded Context



Advanced Topics

- Snapshots
- Command-event conflict resolution
- Eventual consistency
- Transaction-aware repository
- ...

Snapshot



The End

- Eric Evans, Domain-Driven Design
- Greg Young, Unshackle Your Domain