

CQRS Architecture

Agenda

Command Query Responsibility Segregation

History

Core Components

Benefits of CQRS

Common Misconceptions

When to Use CQRS

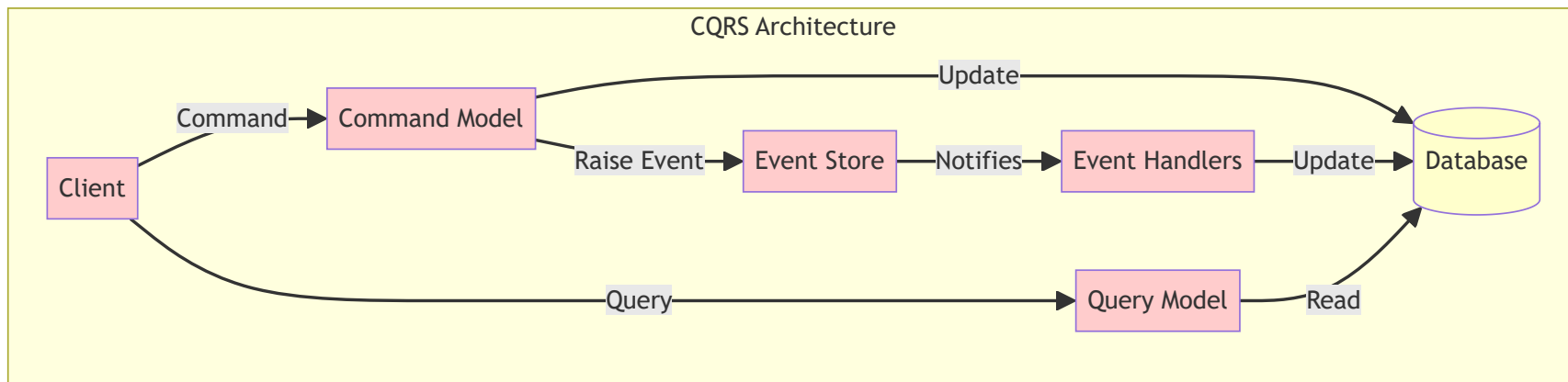
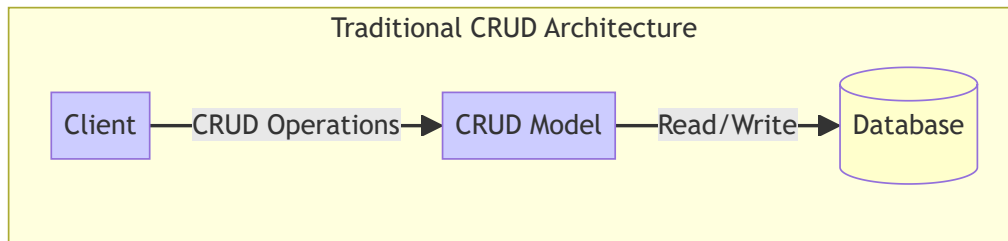
Considerations

References

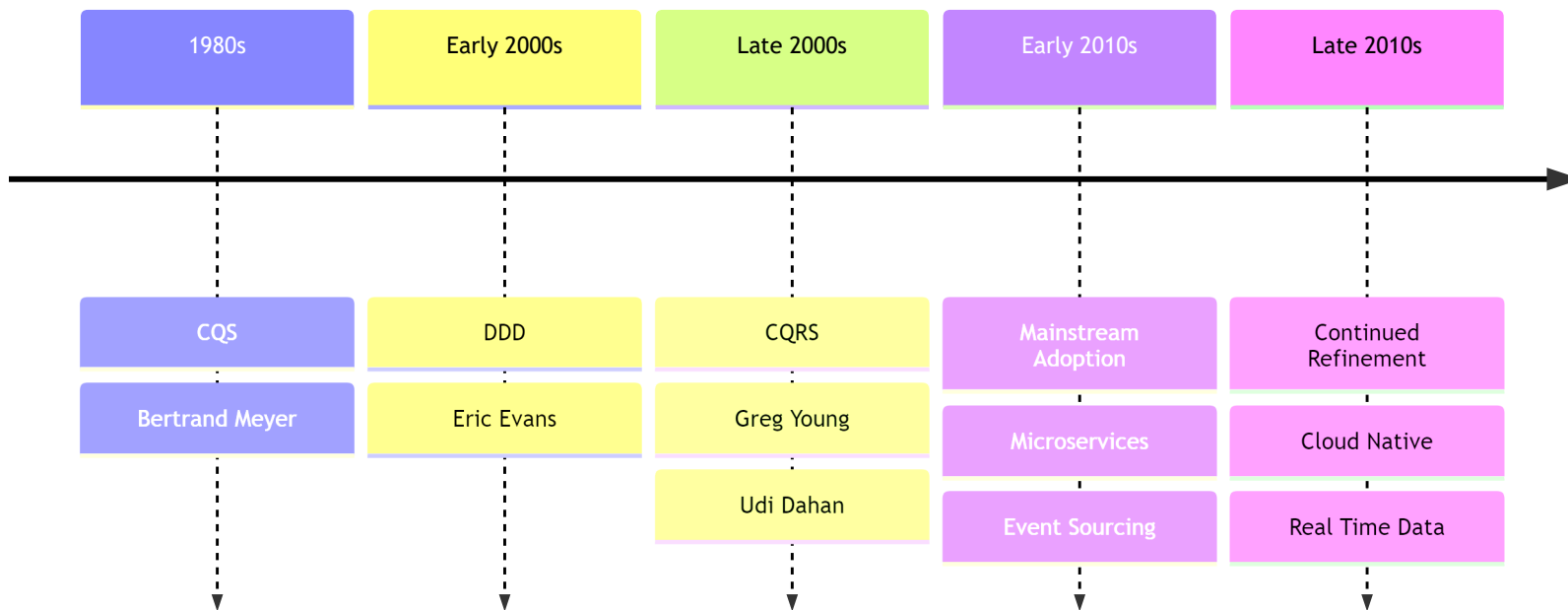
Questions?

Command Query Responsibility Segregation

- What is CQRS?
- How it differs from traditional architectures

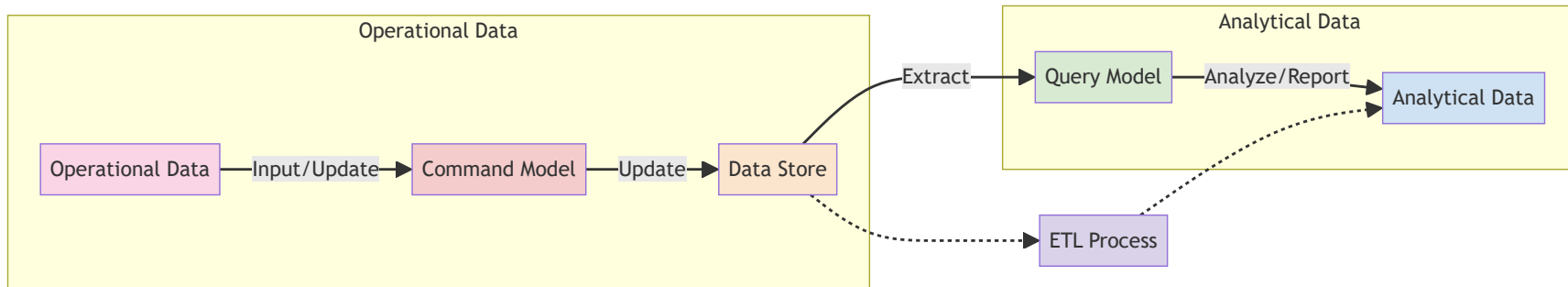


History



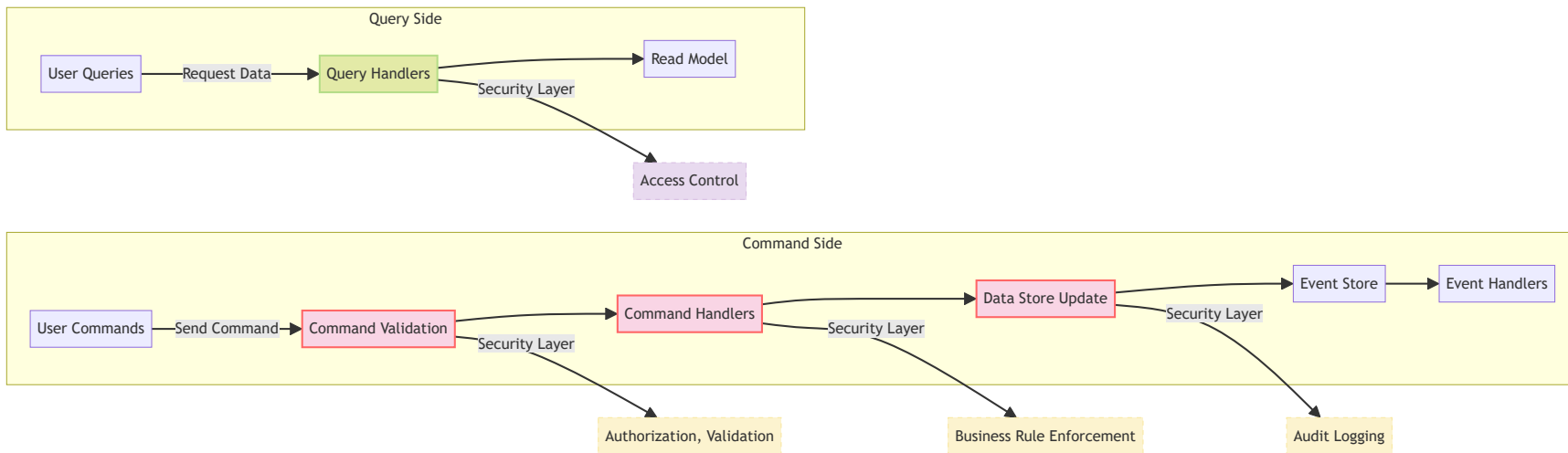
Core Components

- Command and Query Separation
- Operational vs. Analytical data
- Eventual consistency



Benefits of CQRS

- Improved performance
- Scalability
- Simplified query logic
- Security benefits



Common Misconceptions

- It's overly complex
- Only for large systems
- Leads to eventual consistency issues
- Requires Event Sourcing

When to Use CQRS

- Complex domains (bounded contexts in DDD)
- High performance systems
- Systems with clear command-query segregation needs
- Scalable/Distributed Systems
- Systems that required Audit Histories

Considerations

- Initial Complexity
- Steep Learning Curve
- Event Sourcing -> Eventual Consistency

References

- Exploring CQRS and Event Sourcing
- Jason Taylor Clean Architecture
- Ardalis Clean Architecture

Questions?