Cases of different Architecture patterns:

1. **Monolithic Architecture:**
    - When: Small to medium-sized applications with simpler requirements and limited scalability needs.
    - Use Case: Traditional web applications, simple CRUD applications, prototypes.
2. **Microservices Architecture:**
    - When: Complex and large-scale applications with diverse functionalities that need independent scalability and deployment.
    - Use Case: E-commerce platforms, social media networks, complex financial systems.
3. **Serverless Architecture:**
    - When: Event-driven applications with variable workloads, aiming for reduced operational overhead.
    - Use Case: Real-time data processing, IoT applications, event-driven backend services.
4. **Event-Driven Architecture:**
    - When: Applications with asynchronous communication requirements and real-time data processing needs.
    - Use Case: Real-time analytics, IoT data processing, chat applications.
5. **Service-Oriented Architecture (SOA):**
    - When: Applications with a mix of legacy and modern systems requiring interoperability.
    - Use Case: Enterprise applications integrating with multiple third-party systems.
6. **Layered Architecture:**
    - When: Applications with a clear separation of concerns and maintainability as a primary concern.
    - Use Case: Web applications, business applications, content management systems.
7. **Containerization (Docker) Architecture:**
    - When: Applications that need consistent environments across development, testing, and production.
    - Use Case: Microservices, continuous integration and continuous deployment (CI/CD) pipelines.
8. **Event Sourcing and CQRS (Command Query Responsibility Segregation):**
    - When: Applications with complex domain models and high write-to-read ratio.
    - Use Case: Financial systems, e-commerce platforms, systems with audit requirements.
9. **GraphQL Architecture:**

- When: Applications with flexible and efficient data querying needs, and aiming to reduce over-fetching/under-fetching of data.
- Use Case: APIs for mobile apps, dynamic content delivery, applications with varying client data requirements.

10. **Big Data and Lambda Architecture:**
    - When: Applications that need to process and analyze large volumes of data in real-time and batch modes.
    - Use Case: Real-time analytics, IoT data processing, recommendation systems.
11. **Hybrid Architecture:**
    - When: Applications with specific requirements that can't be fulfilled by a single architecture pattern.
    - Use Case: Large-scale applications that combine microservices, serverless, and traditional components based on their needs.