Assignment 1: Resilient Architecture with Cell Based Architecture for E-commerce Company

As a Cloud Architect, you have been tasked to design a resilient architecture for an e-commerce company using the Cell Based Architecture pattern. Your goal is to achieve workload isolation, failure containment, testability, and manageability. Your task is to design a Cell Based Architecture for the given scenarios:

1. High traffic during peak sales season
2. Service disruption due to external factors such as network issues or service provider outage
3. Scaling of services to meet the demand
4. Mitigating risks of a single point of failure
5. Ensuring data security and privacy
6. Providing a seamless experience for customers across multiple devices and platforms

## Questions:

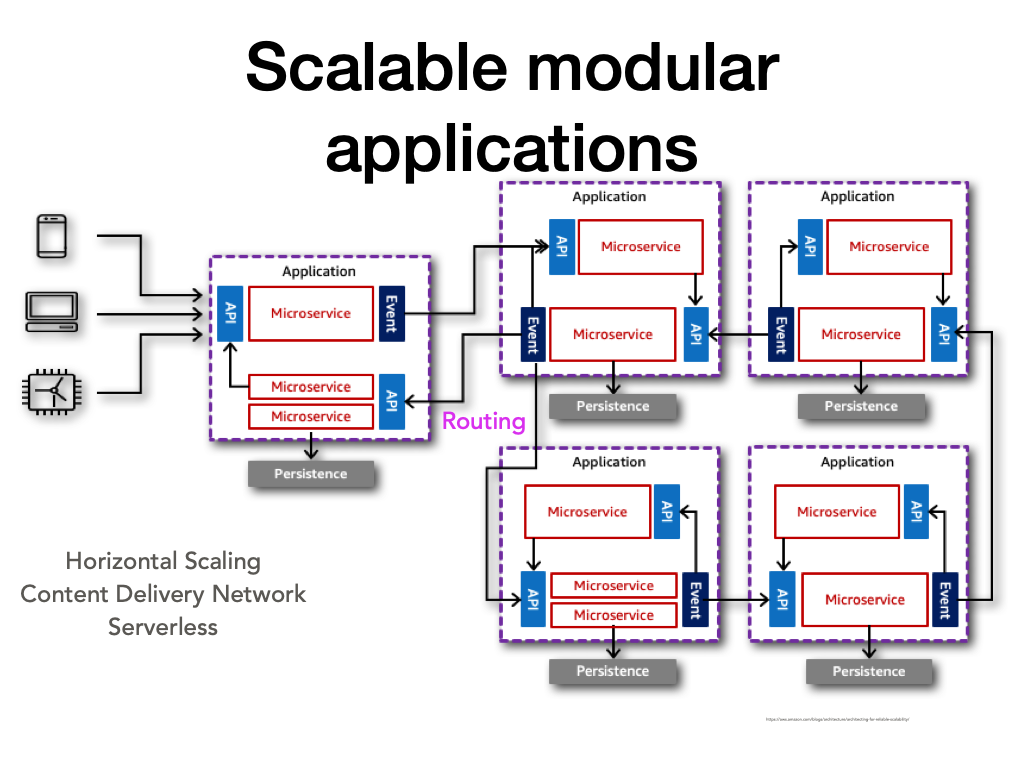
1. What is Cell Based Architecture, and how does it work?
2. What are the benefits of using Cell Based Architecture?
3. How will you ensure workload isolation in the Cell Based Architecture?
4. What are the measures you will take to contain failures in the Cell Based Architecture?
5. How will you ensure testability of the Cell Based Architecture?
6. What are the techniques you will use to ensure manageability of the Cell Based Architecture?

## Checklist:

1. Define the boundaries of the cells
2. Design the service contracts between the cells
3. Choose appropriate communication protocols and messaging patterns between the cells
4. Ensure that each cell is self-contained and can operate independently
5. Define a failure containment strategy for each cell
6. Ensure that each cell has appropriate monitoring and logging mechanisms
7. Design the backup and recovery strategy for the cells
8. Implement security measures such as access control and encryption for data in the cells
9. Ensure that the cells can be tested independently
10. Design the orchestration and coordination mechanisms for the cells.

## Deliverables:

1. A detailed architecture diagram of the Cell Based Architecture pattern for the e-commerce company
2. A written report explaining the benefits of the Cell Based Architecture pattern and how it addresses the given scenarios
3. A list of measures taken to ensure workload isolation, failure containment, testability, and manageability in the Cell Based Architecture pattern.



## Additional Questions:

1. What are the critical services or components in the e-commerce system that require cell-based architecture and bulkhead patterns?
2. What are the specific workload isolation requirements for these critical services or components?
3. What are the specific failure containment requirements for these critical services or components?
4. What are the specific testability requirements for these critical services or components?
5. What are the specific manageability requirements for these critical services or components?

## Checklist:

1. Identify the critical services or components in the e-commerce system that require cell-based architecture and bulkhead patterns.
2. Define the specific workload isolation requirements for each critical service or component.
3. Define the specific failure containment requirements for each critical service or component.
4. Define the specific testability requirements for each critical service or component.
5. Define the specific manageability requirements for each critical service or component.
6. Evaluate the current system architecture to identify any areas that require improvement to meet the cell-based architecture and bulkhead pattern requirements.
7. Implement the necessary changes to the architecture to meet the requirements for cell-based architecture and bulkhead patterns.
8. Test the system to ensure that the cell-based architecture and bulkhead patterns are functioning as expected.
9. Monitor the system to ensure that the cell-based architecture and bulkhead patterns continue to meet the requirements over time.
10. Continuously evaluate the system to identify any new requirements or areas for improvement related to cell-based architecture and bulkhead patterns.

An e-commerce company has a cell-based architecture for their checkout service, which handles the critical task of processing payments and completing orders. The following are some specific examples of how they have implemented the bulkhead pattern to achieve resiliency:

1. Workload isolation: The checkout service is divided into multiple cells, each handling a specific set of tasks. For example, one cell handles payment processing, while another cell handles order confirmation. This ensures that each cell can function independently without affecting the other cells.
2. Failure containment: Each cell has its own resources, such as a dedicated database and cache, which are not shared with other cells. This helps to contain failures to a specific cell and prevent them from cascading to other parts of the system.
3. Testability: The company has implemented automated tests to validate the resilience of their system. They have a suite of tests that simulate failures at different levels of the system, such as network failures, database failures, and hardware failures. These tests are run regularly to ensure that the system can withstand failures and recover quickly.
4. Manageability: The company has implemented a monitoring and alerting system that tracks the health of each cell and alerts the operations team in case of any issues. The system also provides detailed metrics on the performance of each cell, such as response times and error rates, which helps the team to identify and fix issues quickly.

Overall, the cell-based architecture with bulkhead pattern has helped the e-commerce company to achieve high availability and resiliency for their checkout service, which is critical for their business.

## Example:

1. Workload isolation: The checkout service is divided into multiple cells, each handling a specific set of tasks. For example, one cell handles payment processing, while another cell handles order confirmation. This ensures that each cell can function independently without affecting the other cells.
2. Failure containment: Each cell has its own resources, such as a dedicated database and cache, which are not shared with other cells. This helps to contain failures to a specific cell and prevent them from cascading to other parts of the system.
3. Testability: The company has implemented automated tests to validate the resilience of their system. They have a suite of tests that simulate failures at different levels of the system, such as network failures, database failures, and hardware failures. These tests are run regularly to ensure that the system can withstand failures and recover quickly.
4. Manageability: The company has implemented a monitoring and alerting system that tracks the health of each cell and alerts the operations team in case of any issues. The system also provides detailed metrics on the performance of each cell, such as response times and error rates, which helps the team to identify and fix issues quickly.

## Solution:

1. Workload Isolation:

* Identify critical business functions that must be isolated from each other.
* Analyze the impact of an outage in one area on the other areas.
* Evaluate the interdependencies between different workloads.
* Determine the communication patterns between different workloads.
* Evaluate the scalability requirements of different workloads.
* Identify the data protection and compliance requirements of different workloads.

1. Failure Containment:

* Identify the possible failure scenarios and their impact on different workloads.
* Evaluate the effectiveness of existing failure containment measures.
* Identify the gaps in existing failure containment measures.
* Define the required SLAs for recovery time and recovery point objectives.
* Identify the critical data and systems that must be backed up.
* Determine the backup and recovery strategy for different workloads.

1. Testability:

* Identify the critical business functions that must be tested regularly.
* Evaluate the effectiveness of existing testing procedures.
* Identify the gaps in existing testing procedures.
* Define the required testing SLAs for recovery time and recovery point objectives.
* Determine the testing strategy for different workloads.
* Define the test scenarios and test cases for different workloads.
* Define the test automation strategy for different workloads.

1. Manageability:

* Identify the management requirements for different workloads.
* Determine the monitoring and alerting requirements for different workloads.
* Define the metrics and KPIs to measure the performance and availability of different workloads.
* Determine the logging and auditing requirements for different workloads.
* Define the access control and security requirements for different workloads.
* Evaluate the cost of managing different workloads.

Specific Example:

An e-commerce company wants to implement a cell-based architecture to achieve workload isolation, failure containment, testability, and manageability. The company has identified the following workloads:

* Order management
* Inventory management
* Customer management
* Payment management
* Product management
* Workload Isolation:
  + Order management must be isolated from inventory management to prevent inventory errors.
  + Payment management must be isolated from other workloads to prevent unauthorized access to payment information.
  + Customer management must be isolated from other workloads to prevent unauthorized access to customer information.
  + Product management must be isolated from other workloads to prevent unauthorized access to product information.
* Failure Containment:
  + In case of a failure in order management, inventory management, and customer management should not be impacted.
  + In case of a failure in payment management, only payment-related functionality should be impacted.
  + In case of a failure in product management, only product-related functionality should be impacted.
* Testability:
  + Order management, inventory management, and customer management must be tested regularly to ensure data accuracy and system availability.
  + Payment management must be tested regularly to ensure the security of payment information.
  + Product management must be tested regularly to ensure data accuracy and system availability.
* Manageability:
  + Order management, inventory management, and customer management must be monitored and audited regularly to ensure data accuracy and system availability.
  + Payment management must be monitored and audited regularly to ensure the security of payment information.
  + Product management must be monitored and audited regularly to ensure data accuracy and system availability.
  + Metrics and KPIs must be defined to measure the performance and availability of different workloads.
  + Access control and security requirements must be defined for different workloads.