

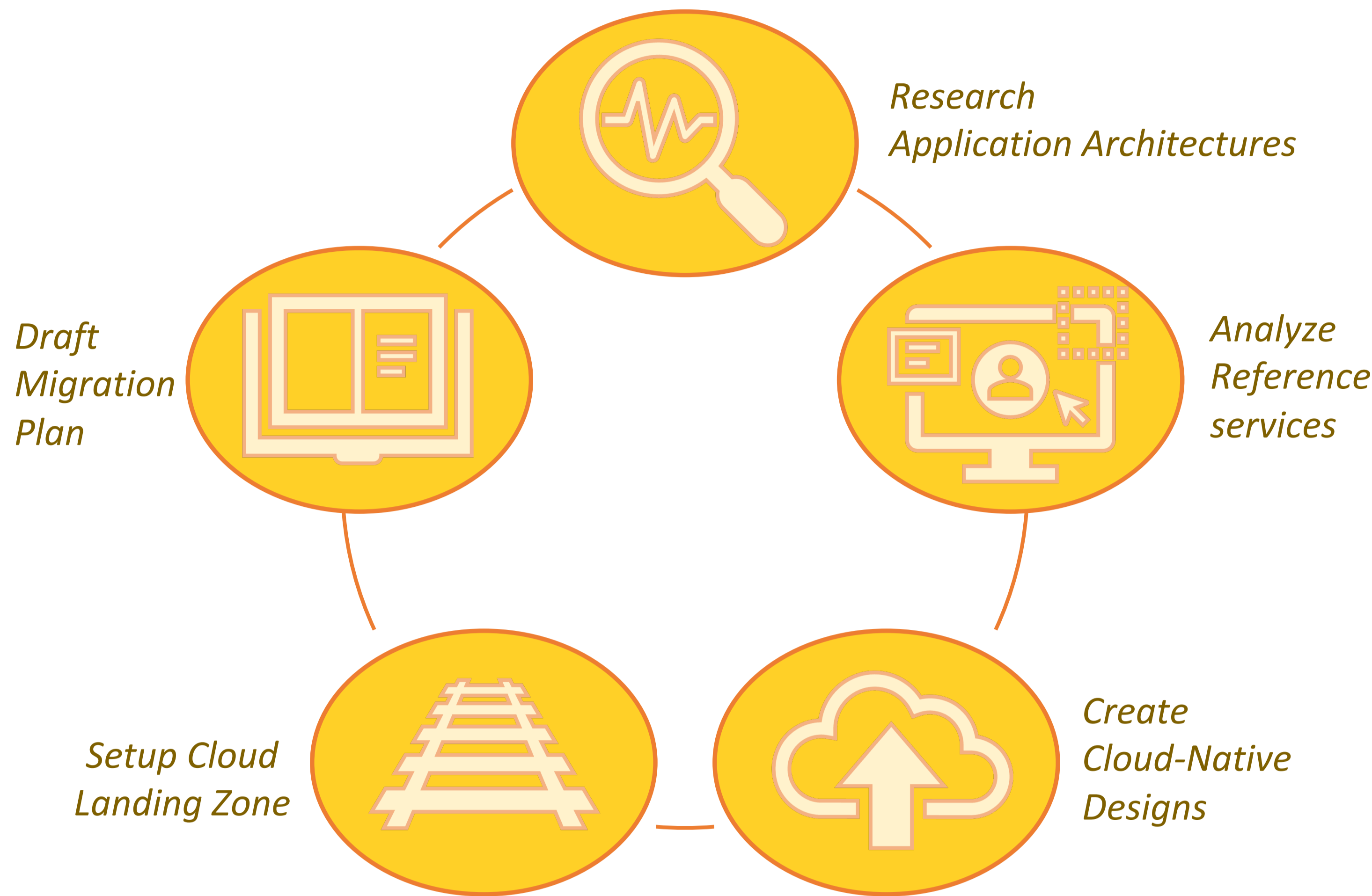
# Development of Application Transformation from on-premise to Cloud

Fahim Satari · Shubham Khetan · Philipp Grotrian · Kyuri Kim · Sujay Chavan · Saakshi Bahrgava · Manoj Nyamaddala · Rizwan Iftikhar · Priyadarshini Mahadeva

## Problem Statement

The problem with on-premise infrastructure, as opposed to cloud solutions, lies in its inherent limitations. On-premise systems often struggle to provide the scalability needed to accommodate fluctuating workloads, lack the cost-efficiency of pay-as-you-go cloud models, and can face challenges in maintaining high availability and disaster recovery capabilities, all of which can hinder business growth and agility.

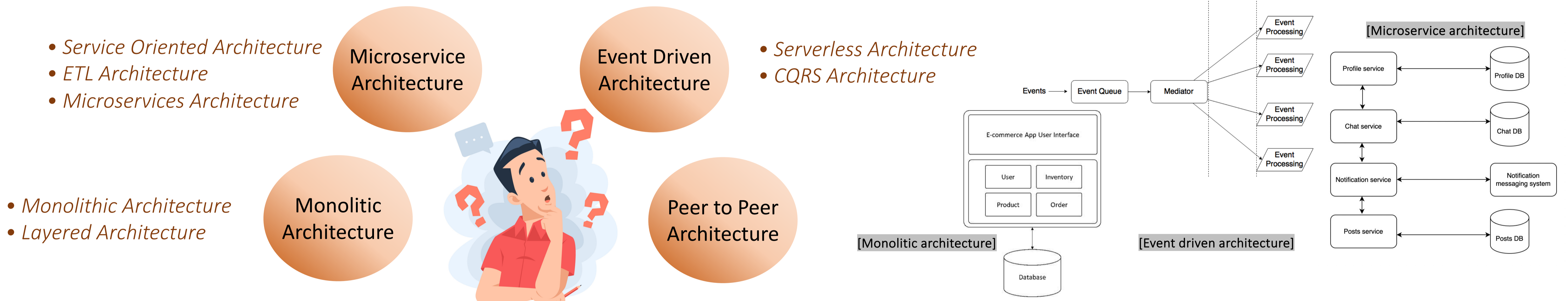
## Tasks



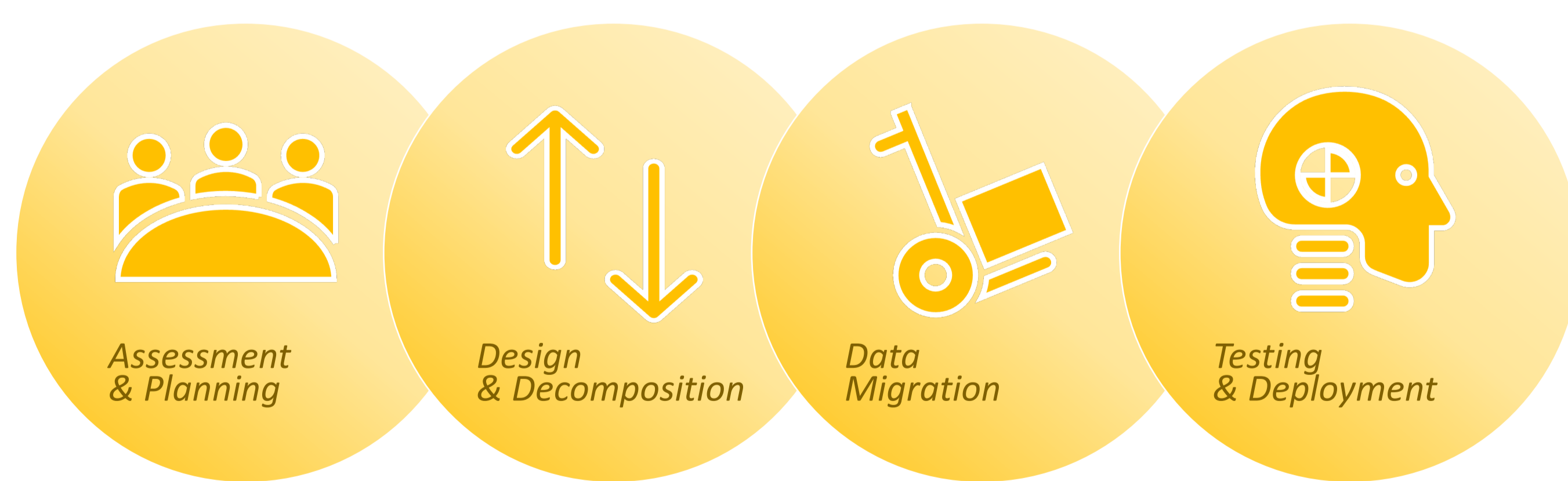
## Researched Architectures

CQRS	Client-Server	ETL
Event-driven Microservice	Event sourcing	Hexagonal
Layered	Microservice	Monolithic
Peer-to-Peer	Serverless	Service-oriented

## Categorization



## Cloud Migration Strategy



Monolithic	Microservices	Peer-to-Peer	Event-Driven
<ul style="list-style-type: none"> <li>Understand current state</li> <li>Define business goals</li> <li>Identify technical requirements</li> <li>Prioritize migration elements</li> </ul>	<ul style="list-style-type: none"> <li>Inventory of current services</li> <li>Define inter-service communication strategies</li> <li>Prioritize services for migration</li> </ul>	<ul style="list-style-type: none"> <li>Evaluate current peer nodes</li> <li>Identify potential bottlenecks</li> </ul>	<ul style="list-style-type: none"> <li>Inventory and assessment of current events</li> <li>Set clear objectives for migration</li> <li>Prefer AWS for scalability and security</li> </ul>
<ul style="list-style-type: none"> <li>Decompose application into manageable pieces</li> <li>Create cloud-native design strategy</li> </ul>	<ul style="list-style-type: none"> <li>Design service boundaries</li> <li>Ensure independent deployability</li> </ul>	<ul style="list-style-type: none"> <li>Design decentralized cloud architecture</li> <li>Ensure data consistency strategies</li> </ul>	<ul style="list-style-type: none"> <li>Design event-driven cloud architecture</li> <li>Ensure low latency and high throughput</li> </ul>
<ul style="list-style-type: none"> <li>Secure data transfer processes</li> <li>Handle large datasets efficiently</li> </ul>	<ul style="list-style-type: none"> <li>Migrate database per service</li> <li>Maintain data integrity across services</li> </ul>	<ul style="list-style-type: none"> <li>Migrate data from each peer</li> <li>Implement data validation checks</li> </ul>	<ul style="list-style-type: none"> <li>Focus on data storage and analytics in the cloud</li> <li>Ensure data consistency across events</li> </ul>
<ul style="list-style-type: none"> <li>Validate each decomposed unit</li> <li>Execute migration strategy</li> <li>Implement post-migration support</li> </ul>	<ul style="list-style-type: none"> <li>Validate each service independently</li> <li>Monitor inter-service communications</li> </ul>	<ul style="list-style-type: none"> <li>Validate decentralized operations</li> <li>Monitor peer communications in the cloud</li> </ul>	<ul style="list-style-type: none"> <li>Integrate and test event-driven components</li> <li>Optimize event processing in the cloud</li> </ul>

## Review and Summary

### Analysis of Architectures

Understood the existing application architectures

### Categorization of Architectures

Analyzed requirements and classified the architectures including cost and security

### Cloud Deployment Models

Identified the appropriate cloud deployment models

### Migration Strategy

Developed Comprehensive migration strategy, including risk assessment and resource planning