MP1: Re-host
**MP1: Re-host**

**Definition:** An application (component) is re-hosted as-is on cloud platform(s)

**Problem:** Resource constraints limit scalability, Need to improve the SLO, Single point of failure, Reduce cost of ownership, Modernization strategies

**Solution:** Re-host on cloud environments, make use of elastic resources, multiple cloud deployment for failover and scalability.

**Benefits:** Improved Backup and Failover, Coarse-grained scalability at application level, Simple coarse-grained re-deployment.

**Risks:** Existing architecture constrains portability, deployment time and cost, scalability, integration may introduce greater complexity.
MP2: Cloudification
MP2: Cloudification

Definition: An application (component) is hosted on premise as-is but use public cloud services for extending capabilities instead of on premise components.

Problem: Need to improve reusability, extensibility, Avoid reinventing the wheel by consuming existing publicly accessible cloud services

Solution: Extend the on premise application by integrating with existing public cloud services.

Benefits: Improved time to market.

Risks: Integration may introduce greater complexity.
MP3: Relocation and Optimization
MP3: Relocation and Optimization

**Definition:** A component re-hosted (or relocated) on a cloud platform is optimized but without evolution in the application architecture.

**Problem:** The performance of an application needs to be enhanced without the significant effort of architecture change, and without incurring capital expenditure for on-premise hardware.

**Solution:** Leverage cloud platform services (e.g., IaaS, PaaS services) to improve throughput by leveraging Queue, Database partitioning & sharding, NoSQL, Cache

**Benefits:** As component re-hosting in cloud and optimized performance.

**Risks:** The type of application requests changes over time for example proportion of read only calls reduces, Cloud provider does not provide the necessary services to wrap the optimizations around the application without re-architecting.
MP4: Multi-Cloud Relocation

Before Migration

C1

Cloud Platform X

Cloud Platform Y

On-premise Platform

After Migration

C1

Cloud Service

Cloud Platform X

Cloud Platform Y

On-premise Platform
MP4: Multi-Cloud Relocation

**Definition**: A component re-hosted (or relocated) on a cloud platform is enhanced by using the environmental services of the other cloud platforms.

**Problem**: The availability of an application needs to be enhanced without the significant effort of architecture change, and without incurring capital expenditure for on premise hardware.

**Solution**: Leverage cloud platform environment services to improve availability, e.g., live migration from existing platform to the target platform in case of service outage.

**Benefits**: As component re-hosting in multiple cloud platforms and improve availability and avoid vendor lock-in.

**Risks**: Cloud providers does not provide the necessary services to enable application to run in multiple cloud platforms without re-architecting or rewriting the code.
MP5: Multi-Cloud Refactor

Before Migration

C1

C2

Cloud Service

Cloud Service

Cloud Platform X

Cloud Service

Cloud Platform Y

On-premise Platform

After Migration

C2

C1

Cloud Service

Cloud Service

Cloud Platform X

Cloud Platform Y

On-premise Platform

Cloud Service
Definition: An on premise application is re-architected for deployment on cloud platform to provide better QoS.

Problem: Coarse-grained applications are not agile enough to respond to requirement changes or variations in workload, and cannot take full advantage of the SLO improvements that can be offered by cloud platforms.

Solution: The application is re-architected as a set of fine-grained components, The deployment of high-usage components can be optimized independently of low-usage ones, Parallel design for better throughput to multiple cloud platforms, Components designed as independent integrity units to reduce dependencies and enable replacement.

Benefits: Optimal scalability and performance, wider range of multi-cloud deployment options, agility to respond to business and IT change.

Risks: On premise application is modernized in isolation, and not as part of a portfolio that ensures consistent architecture, Modernization is performed primarily for technical reasons resulting in continued sub-optimal response to business change, Component architecture is only determined bottom-up from existing APIs, Transaction and data integrity approaches may need to be re-evaluated because of multi-cloud environment.
MP6: Hybrid Refactor

Before Migration

C1

C2

Cloud Platform X

On-premise Platform

Cloud Platform Y

After Migration

C1

C2

Cloud Platform X

On-premise Platform

Cloud Platform Y
MP7: Hybrid Refactor with On Premise Adaptation

Before Migration

C1
Cloud Service
On-premise Platform
Cloud Platform X
Cloud Service
Cloud Platform Y

C2
Cloud Service

After Migration

C1
A1
Cloud Service
On-premise Platform
Cloud Platform X

C2
Cloud Service

Cloud Service
Cloud Platform Y
**MP7: Refactor with On Premise Adaptation**

**Definition:** A re-architected application is deployed partially on a cloud environment and partially to its current on premise platform.

**Problem:** All components of the re-architected application may not be suitable for deployment on cloud. For example, due to sensitivity of data, lack of cloud capability to support current feature of application, license restrictions, or to support a gradual migration plan. However, as they are not co-located, some mechanisms are required to integrate the components.

**Solution:** A component adapter (e.g., on premise façade) is adopted to provide integration of the on premise components with re-hosted cloud-based components.

**Benefits:** Sensitive data remains isolated and in-house

**Risks:** Integration is dependent on specific cloud platform and may cause vendor lock-in.
MP8: Hybrid Refactor with Cloud Adaptation
MP8: Refactor with Cloud Adaptation

Definition: An interface is implemented to provide loose-coupled access to components re-hosted on cloud platform.

Problem: Components re-hosted as-is lacks appropriate service interfaces for integration

Solution: Build a façade, hosted in the cloud platform

Benefits: Loose coupling, platform independent interoperability

Risks: Lack of suitable API on legacy application, Existing application process may not compliant with message-based interaction as a common style in the cloud. Façade is not provided as part of a well-formed service architecture.
MP9: Hybrid Refactor with Hybrid Adaptations
MP10: Multi-Cloud Rebinding
MP10: Multi-Cloud Rebinding

**Definition:** A re-architected application is deployed partially on multiple cloud environments and enables the application to continue to function using secondary deployment when there is a failure with the primary platform.

**Problem:** A natural disaster, such as a hurricane, or a failure such as a software bug or configuration error that impacts cloud services may cause a failure to a cloud platform and stop that from functioning.

**Solution:** The same architecture for resilient systems that route users to the closest data center can be used to account for failover scenarios. In particular, they can be configured to monitor the health of the services to which they are directing users, and if any service is unavailable, traffic will be routed to a healthy instance. An on-premise component adapter (e.g., service bus or elastic load balancer) is adopted to provide integration of the components in different cloud platforms.

**Benefits:** As unhealthy services become healthy again, traffic can be delivered, returning system responsiveness to maximum levels.

**Risks:** This scheme does not guarantee instant or seamless failover. There will be downtime.
MP11: Multi-Cloud Rebinding with Cloud Brokerage
MP12: Replacement

Before Migration

C1

C2

Cloud Platform X

Cloud Platform Y

On-premise Platform

Cloud Service

Cloud Service

After Migration

C1

On-premise Platform

Cloud Platform X

Cloud Platform Y

Cloud Service

Cloud Service
MP12: Replacement

**Definition:** Individual capabilities in a re-architected solution are re-provisioned rather than re-engineered.

**Problem:** Some of existing components provided by the current application are not the best alternative to meet business requirements.

**Solution:** Analyze the requirements and identify a set of capabilities that can be replaced by existing cloud services. The provisioning of each capability is assessed by considering current systems analysis on the existing application to identify which of these capabilities could be supported by the re-architected system, alternative cloud services that provide a benefit over the re-engineering of the current capability are identified and replace existing components.

**Benefits:** The solution is improved though best-in-class cloud services, Re-engineering costs and effort are saved.

**Risks:** Cloud services presume specific communication protocol that make the replacement a challenging tasks.
MP13: Replacement with On Premise Adaptation

Before Migration

C1

C2

On-premise Platform

Cloud Platform X

Cloud Platform Y

After Migration

C1

A1

On-premise Platform

Cloud Platform X

Cloud Platform Y

Cloud Service

Cloud Service

Cloud Service

Cloud Service

Cloud Service
MP14: Replacement with Cloud Adaptation
MP15: Multi-application Modernization
Definition: On premise applications are re-architected as a portfolio and deployed on cloud environment.

Problem: The re-architecting of on premise applications in isolation does not remove inconsistencies in data or duplicated functionalities, nor reduce the cost of their combined operation or maintenance.

Solution: Current applications are analyzed as a portfolio to identify opportunities for consolidation and sharing. The separation of the service architecture and the solution architecture enables the identification of components (capabilities) that are shared by more than one solution.

Benefits: Consistent information and rules in shared components, Reduced operation and maintenance costs for shared components, Foundation for more agile delivery of subsequent new applications.

Risks: Lack of business commitment to shared capabilities.