

# Amazon ElastiCache

# Table of Contents



1. What is Amazon ElastiCache?
2. ElastiCache Use Cases
3. ElastiCache Serverless
4. Designing Custom ElastiCache Clusters
5. How ElastiCache Serverless Works
6. How ElastiCache self-designed clusters Works
7. ElastiCache Processing Units (ECPUs)

# What is Amazon ElastiCache?

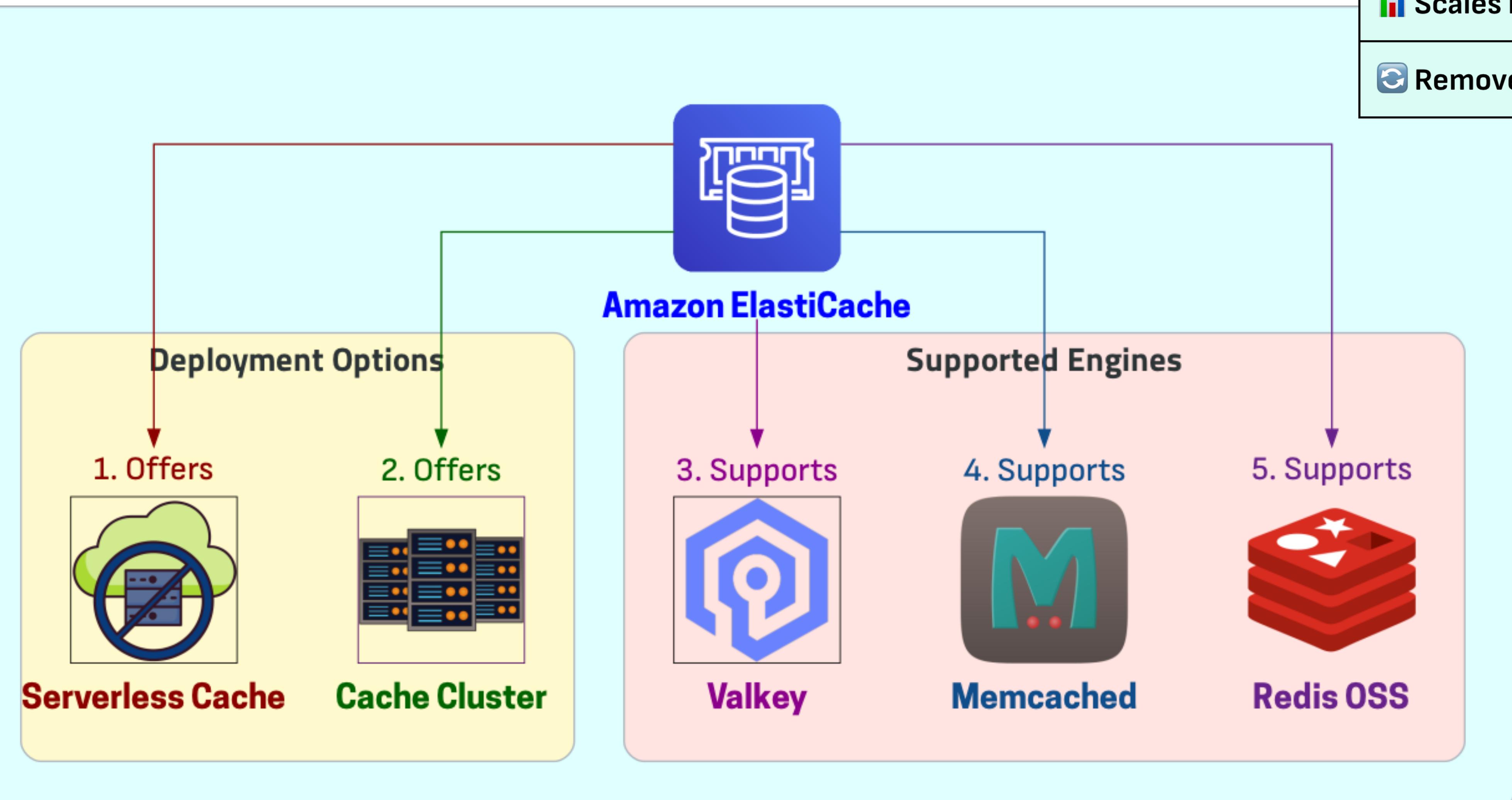
1. 🚀 Cloud Caching Service

Web service for distributed caching

Simplifies setup and management

Scales in-memory data stores

Removes complexity



# What is Amazon ElastiCache?

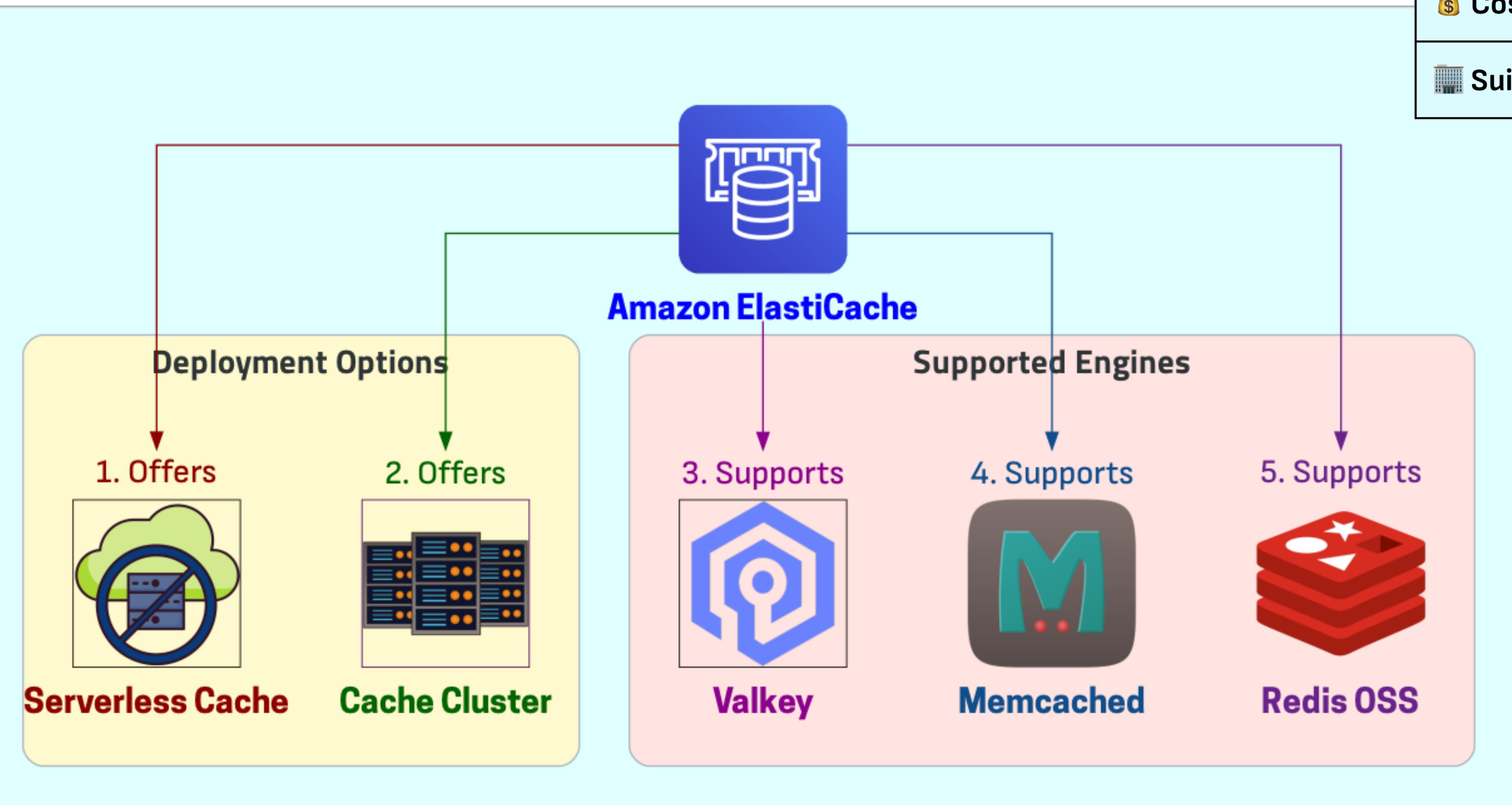
## 2. ⚡ Performance Benefits

⚡ High-performance capabilities

📈 Scalability features

💰 Cost-effective solution

🏢 Suitable for organizations



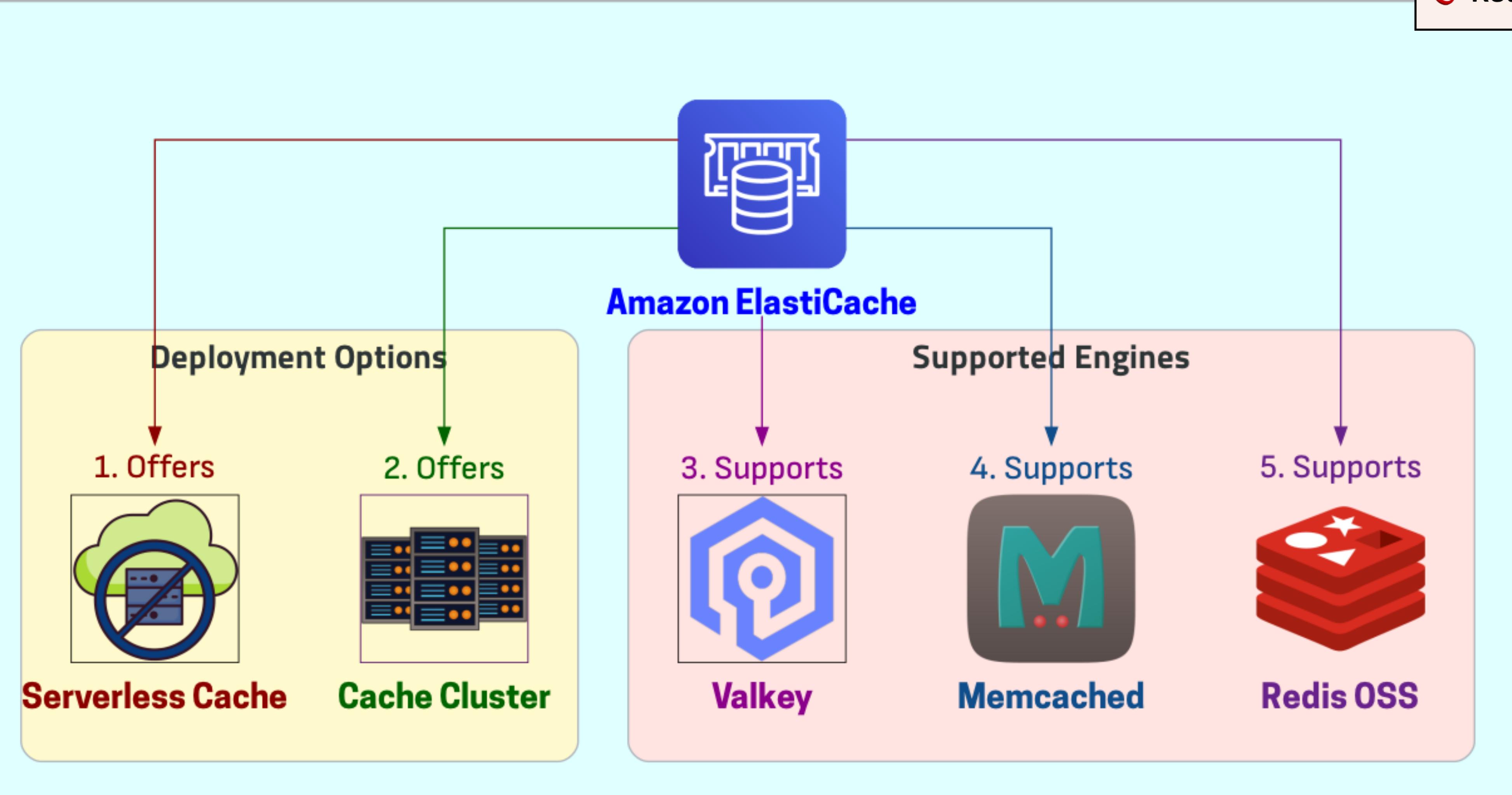
# What is Amazon ElastiCache?

## 3. Deployment Options

Cloud Serverless cache option

Custom cache cluster

Requirement-based choice



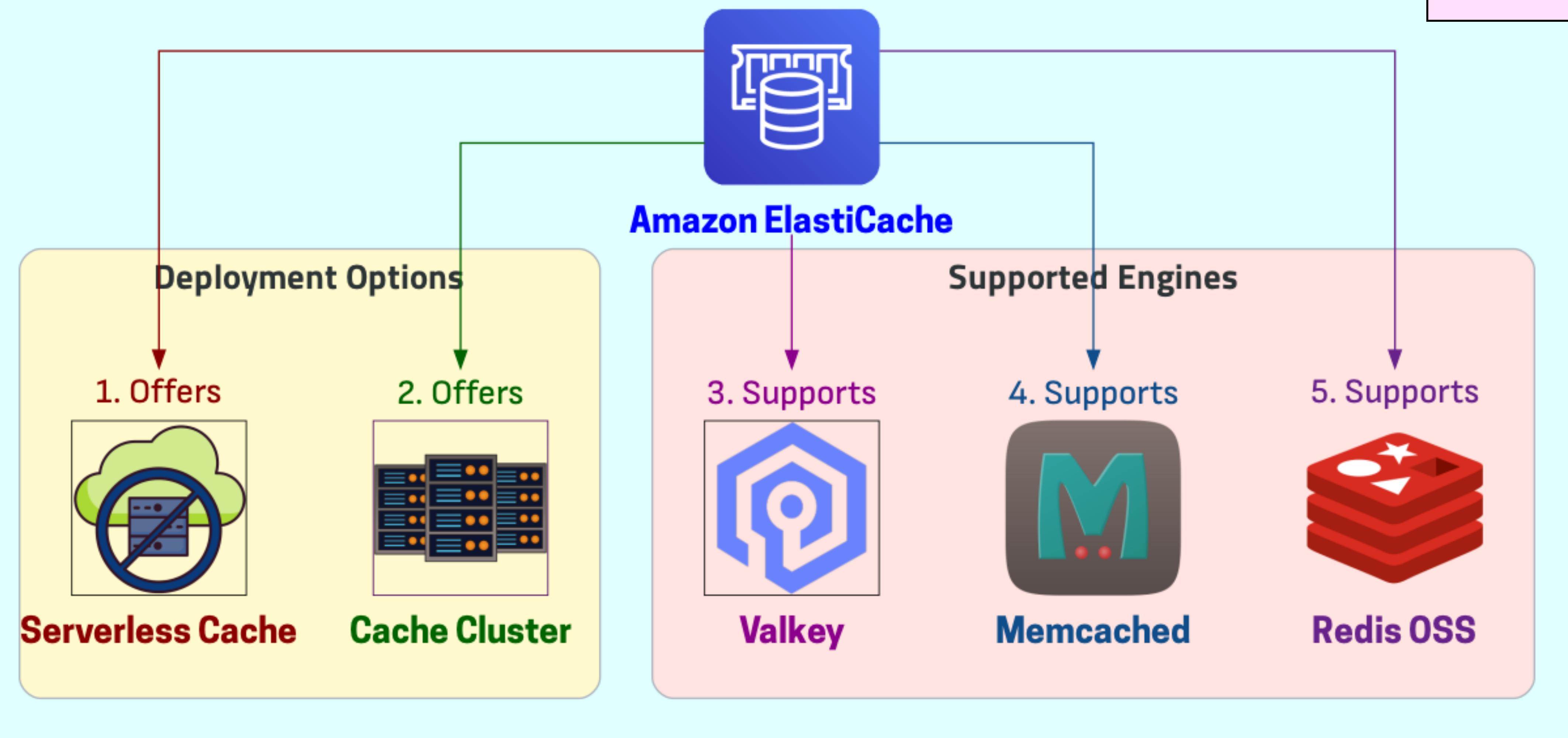
# What is Amazon ElastiCache?

4. Engine Compatibility

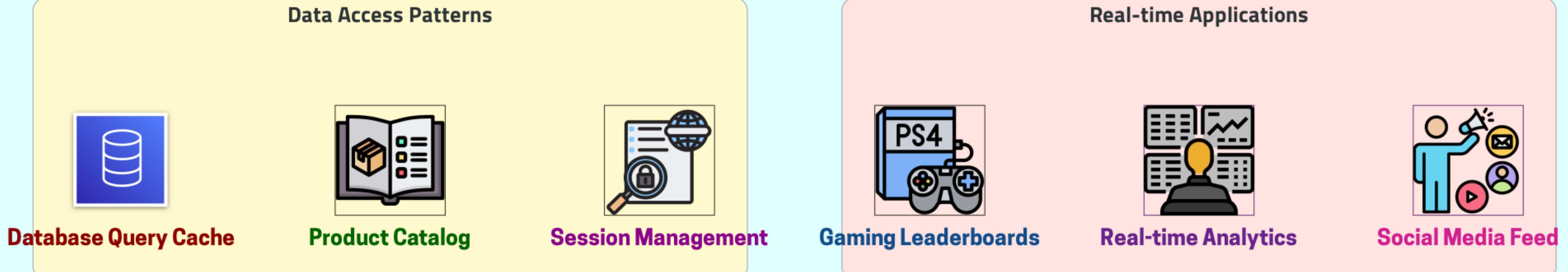
Valkey support

Memcached support

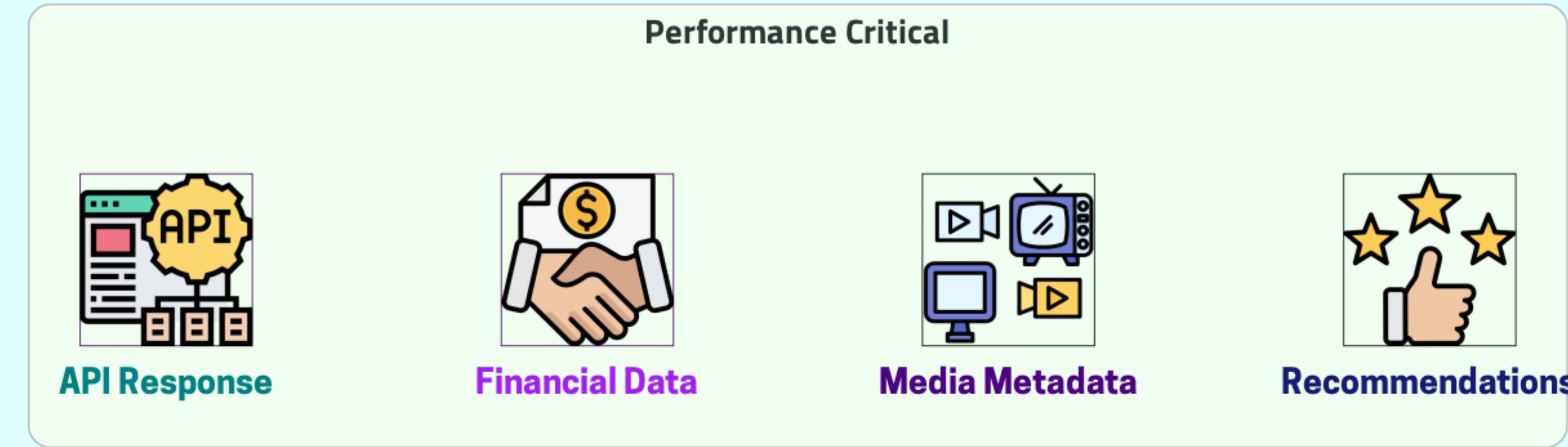
Redis OSS support



# ElastiCache Use Cases



ElastiCache



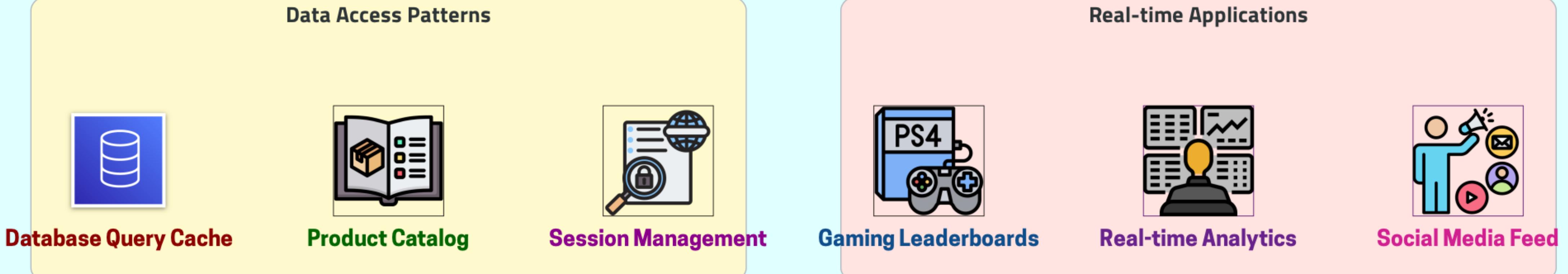
## 1. Database Query Caching

⚡ Improves response times

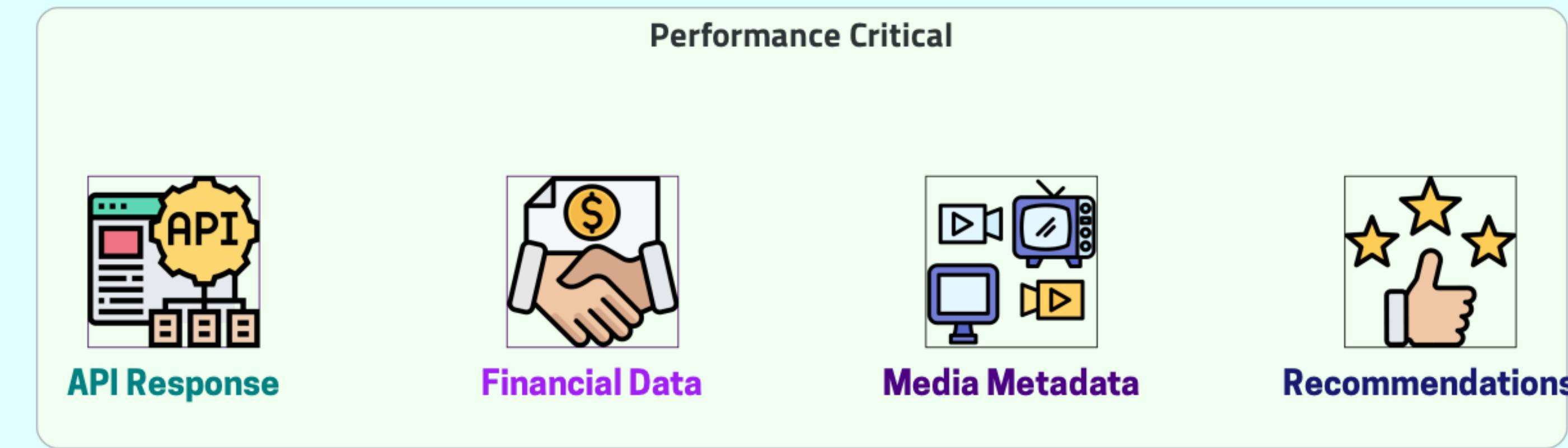
⬇ Reduces database load

🚀 Enhanced performance

# ElastiCache Use Cases



ElastiCache



2. 🛍️ E-commerce Product Catalog

124 High-traffic retail sites

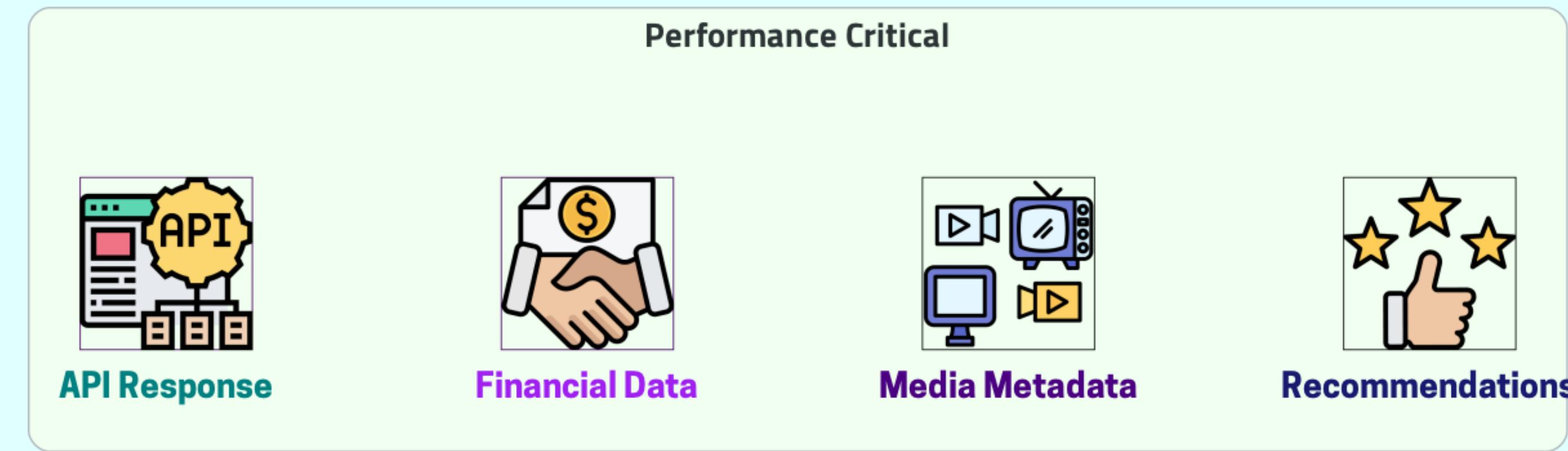
💰 Real-time pricing updates

📊 Peak event handling

# ElastiCache Use Cases



ElastiCache



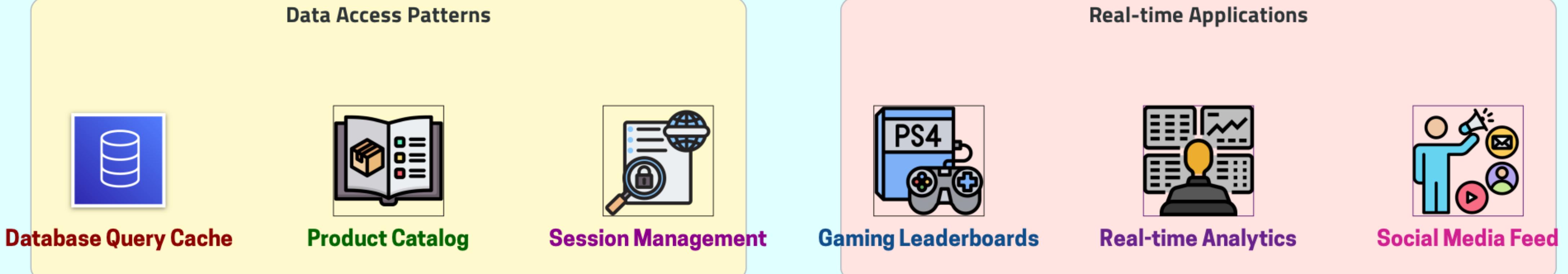
3. Session State Management

User state persistence

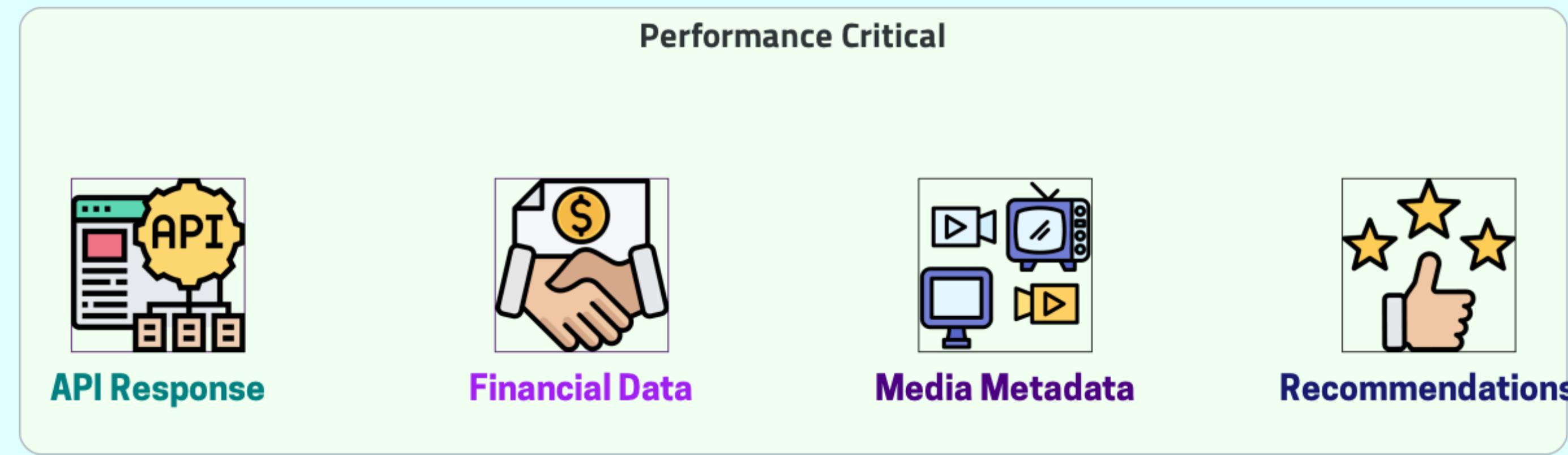
Web application support

State synchronization

# ElastiCache Use Cases



ElastiCache



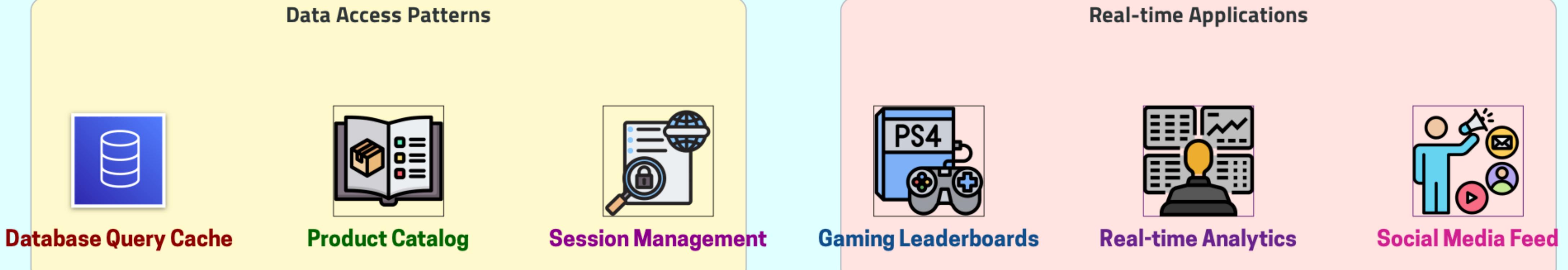
4. Gaming Leaderboards

⚡ Sub-millisecond latency

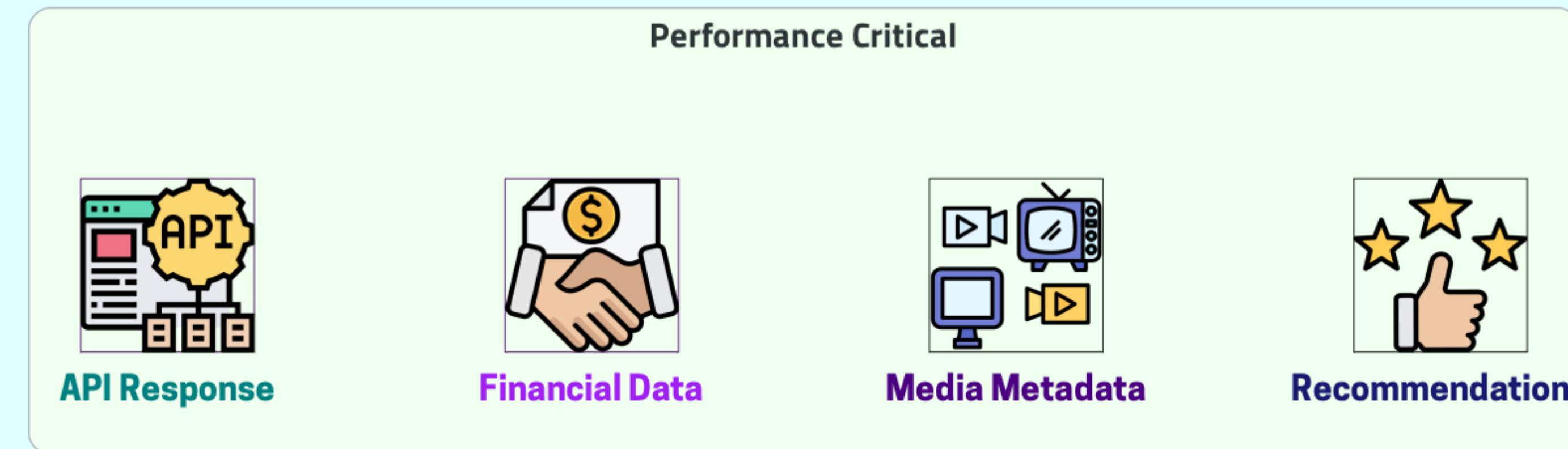
🏆 Real-time rankings

🎯 Instant updates

# ElastiCache Use Cases



ElastiCache



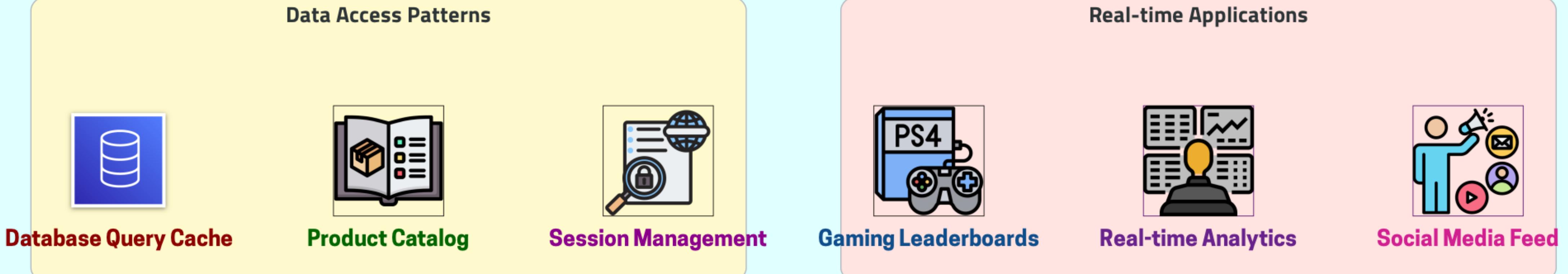
5. **Real-time Analytics**

**Business intelligence**

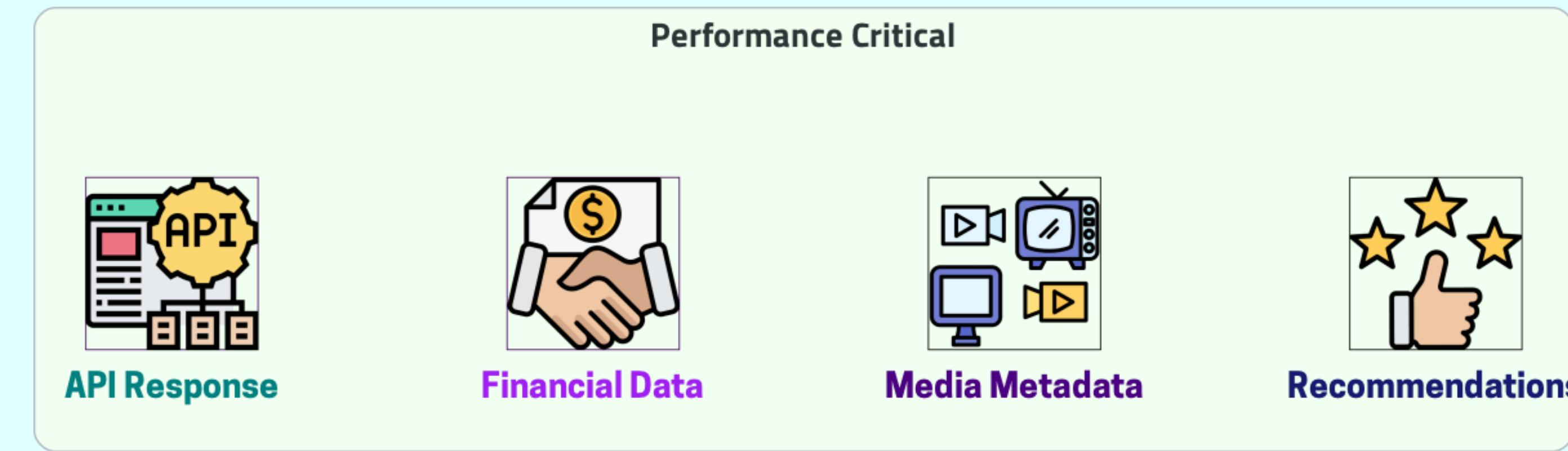
**User behavior tracking**

**Metrics monitoring**

# ElastiCache Use Cases

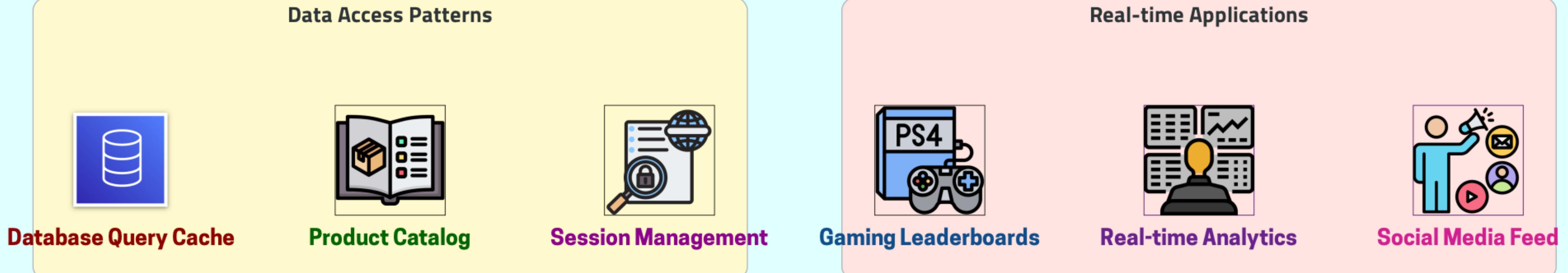


ElastiCache

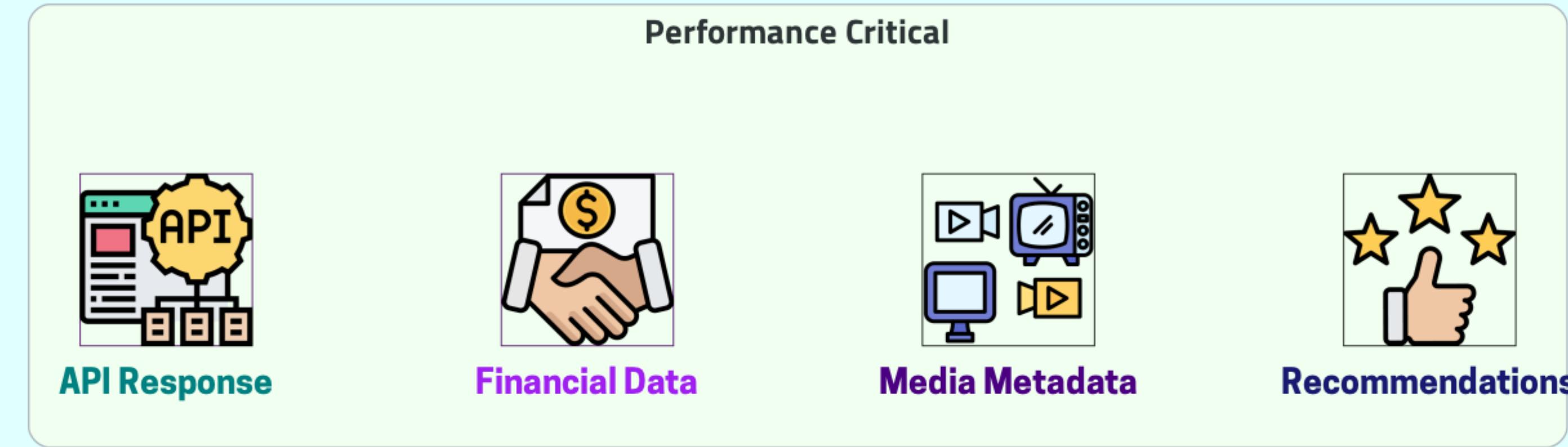


- 6. Social Media Feed
- Instant updates
- Real-time interactions
- Platform responsiveness

# ElastiCache Use Cases



ElastiCache



## 7. API Response Caching

⚡ High-performance APIs

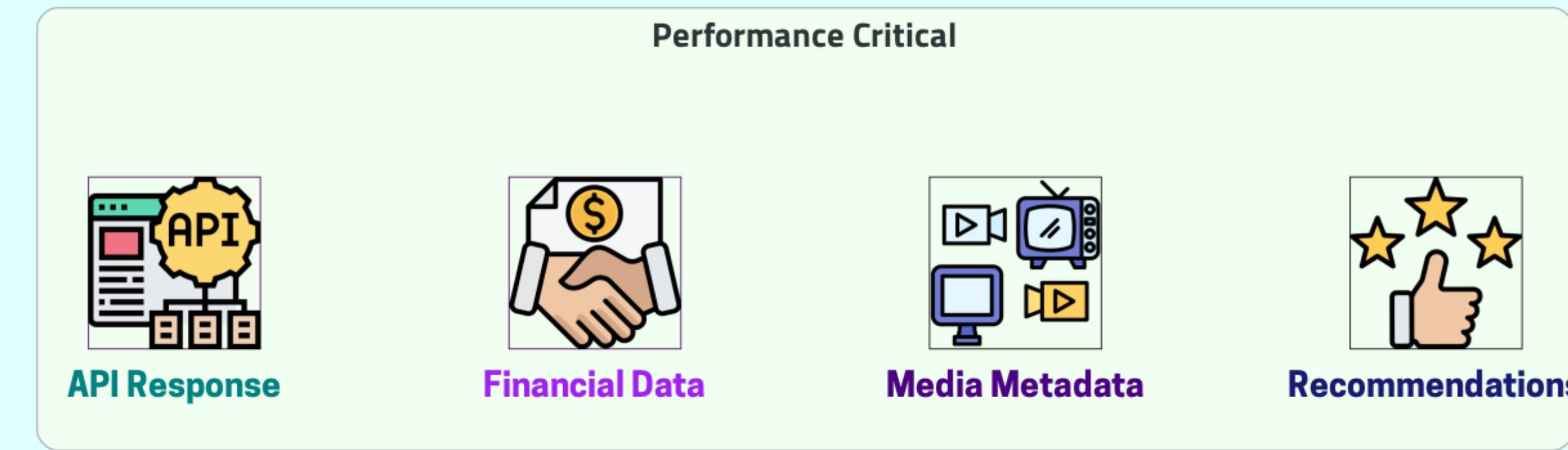
⬇ Reduced backend load

🚀 Improved response times

# ElastiCache Use Cases



ElastiCache



8. Financial Market Data

Time-sensitive operations

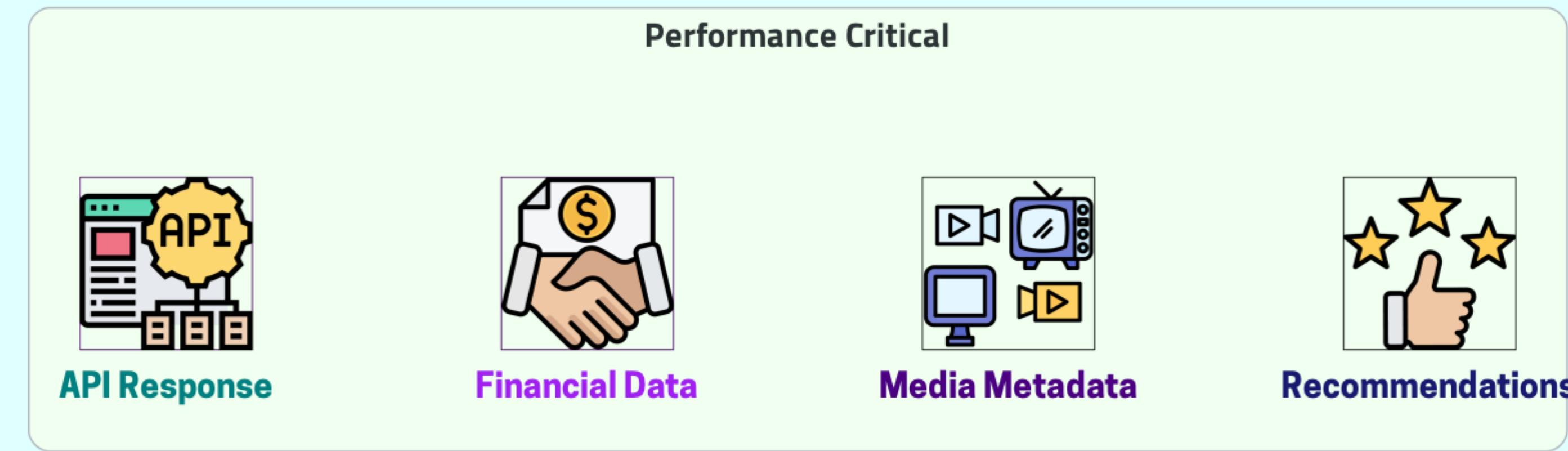
Low latency requirements

Real-time market data

# ElastiCache Use Cases



ElastiCache



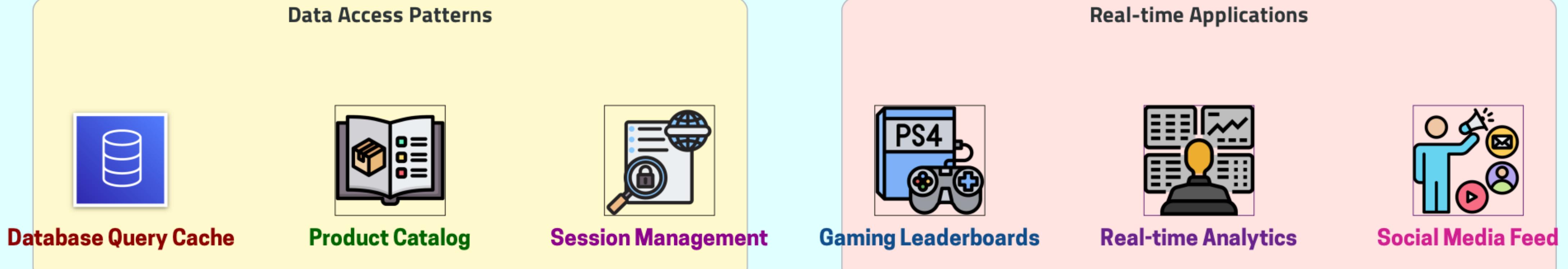
9. Media Streaming Metadata

Smooth content delivery

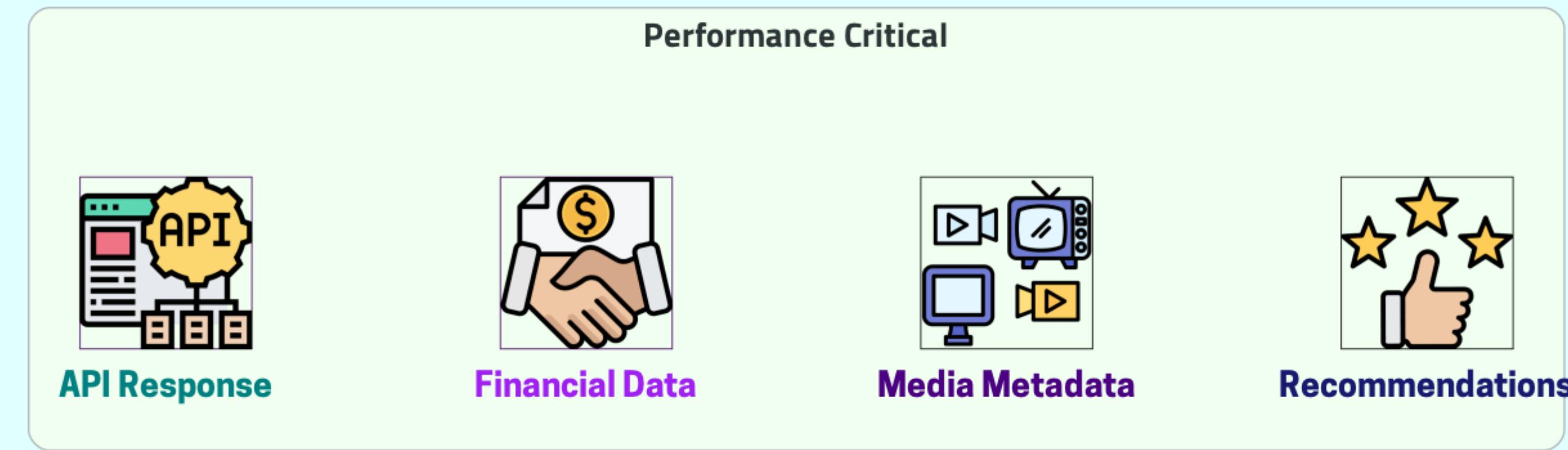
Playback optimization

Streaming enhancement

# ElastiCache Use Cases



ElastiCache



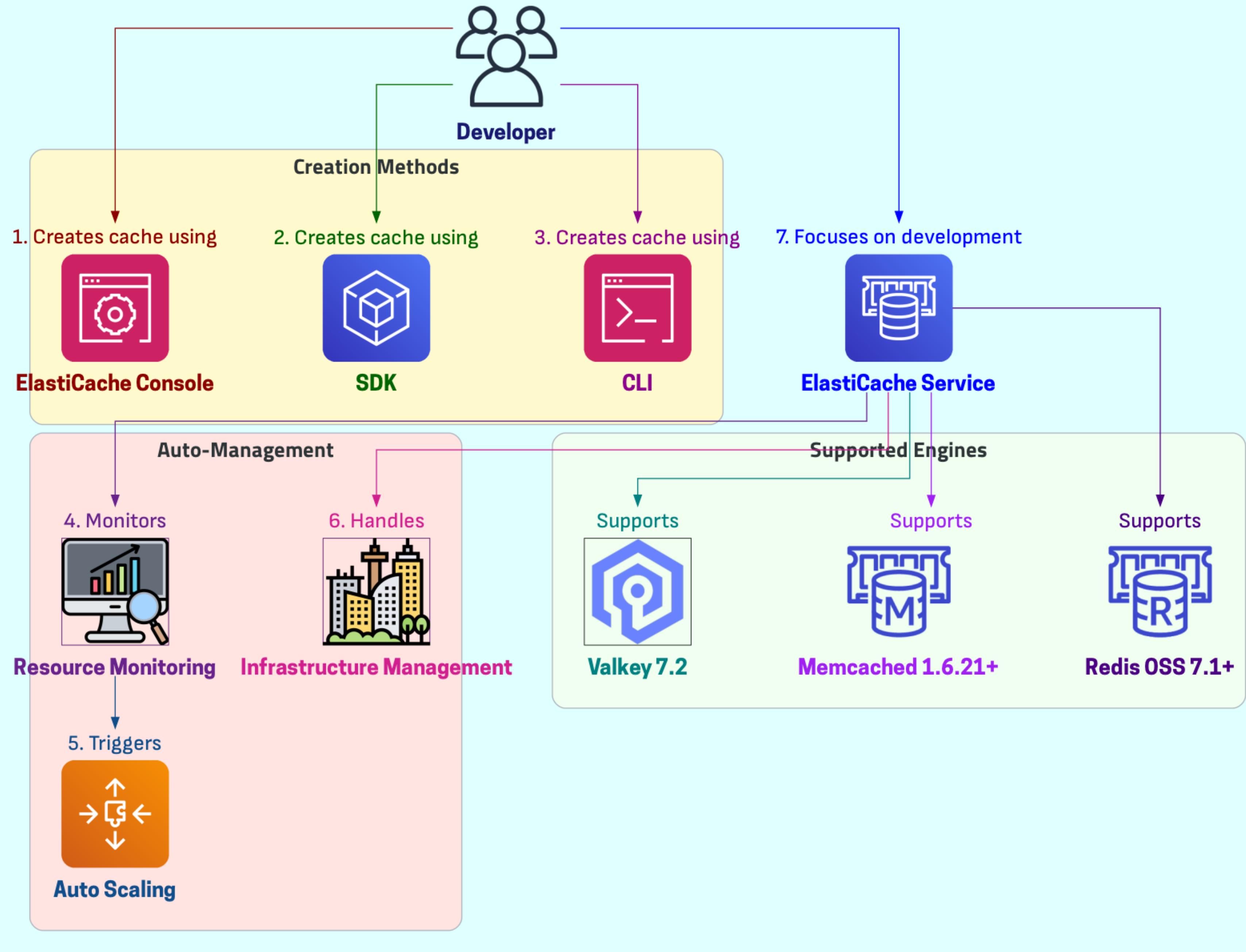
10. **Recommendation Engine**

**Personalized experiences**

**Custom suggestions**

**Modern application support**

# ElastiCache Serverless



## 1. 🚀 Quick Setup

⚡ Under 1-minute deployment

⟳ No provisioning needed

📦 No cluster configuration

## 2. 🔧 Multiple Creation Methods

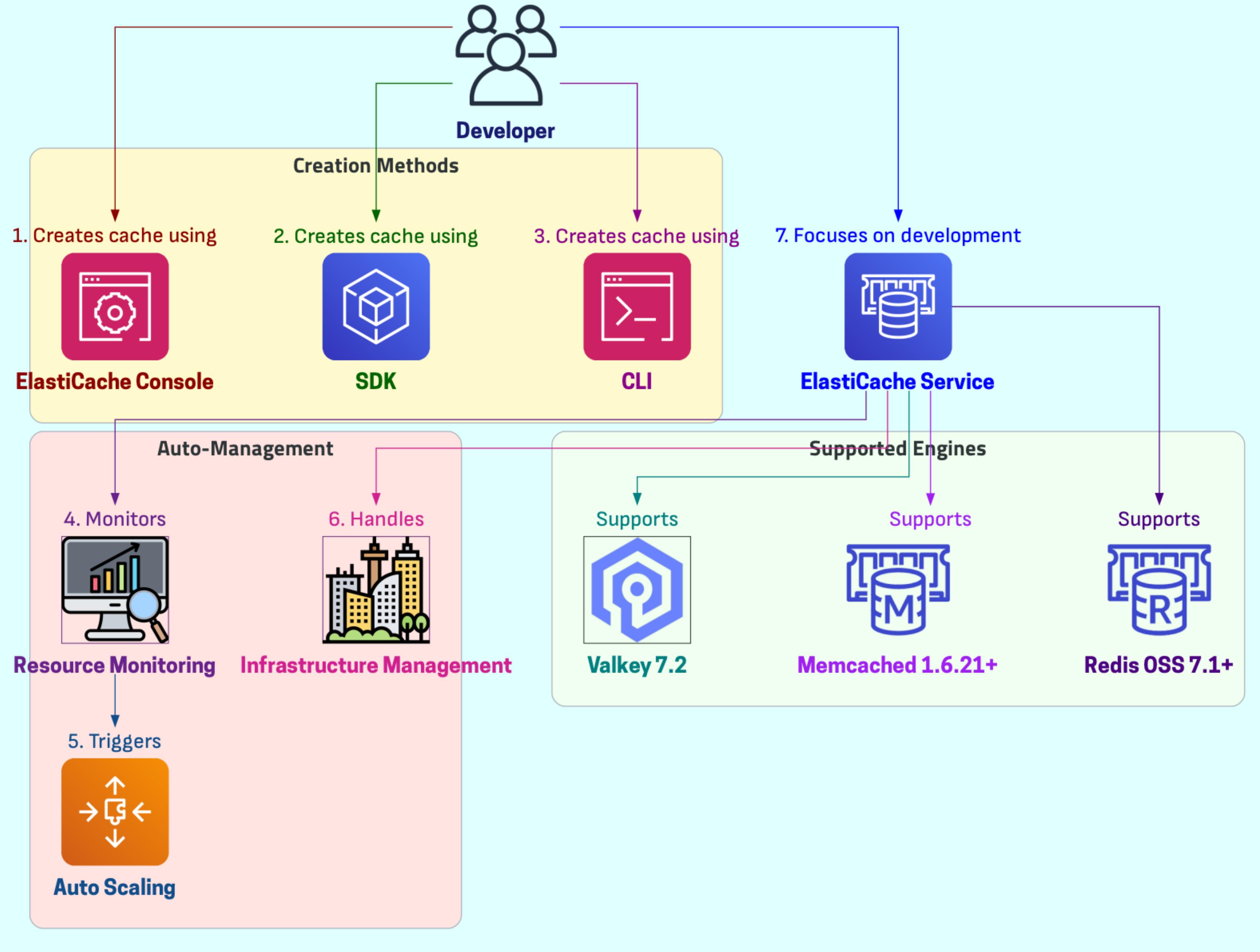
💻 ElastiCache console

📱 SDK integration

⌨️ CLI access

🏷️ Simple cache naming

# ElastiCache Serverless



## 3. Automated Management

### Resource monitoring

Memory usage

Compute resources

Network bandwidth

### Auto-scaling

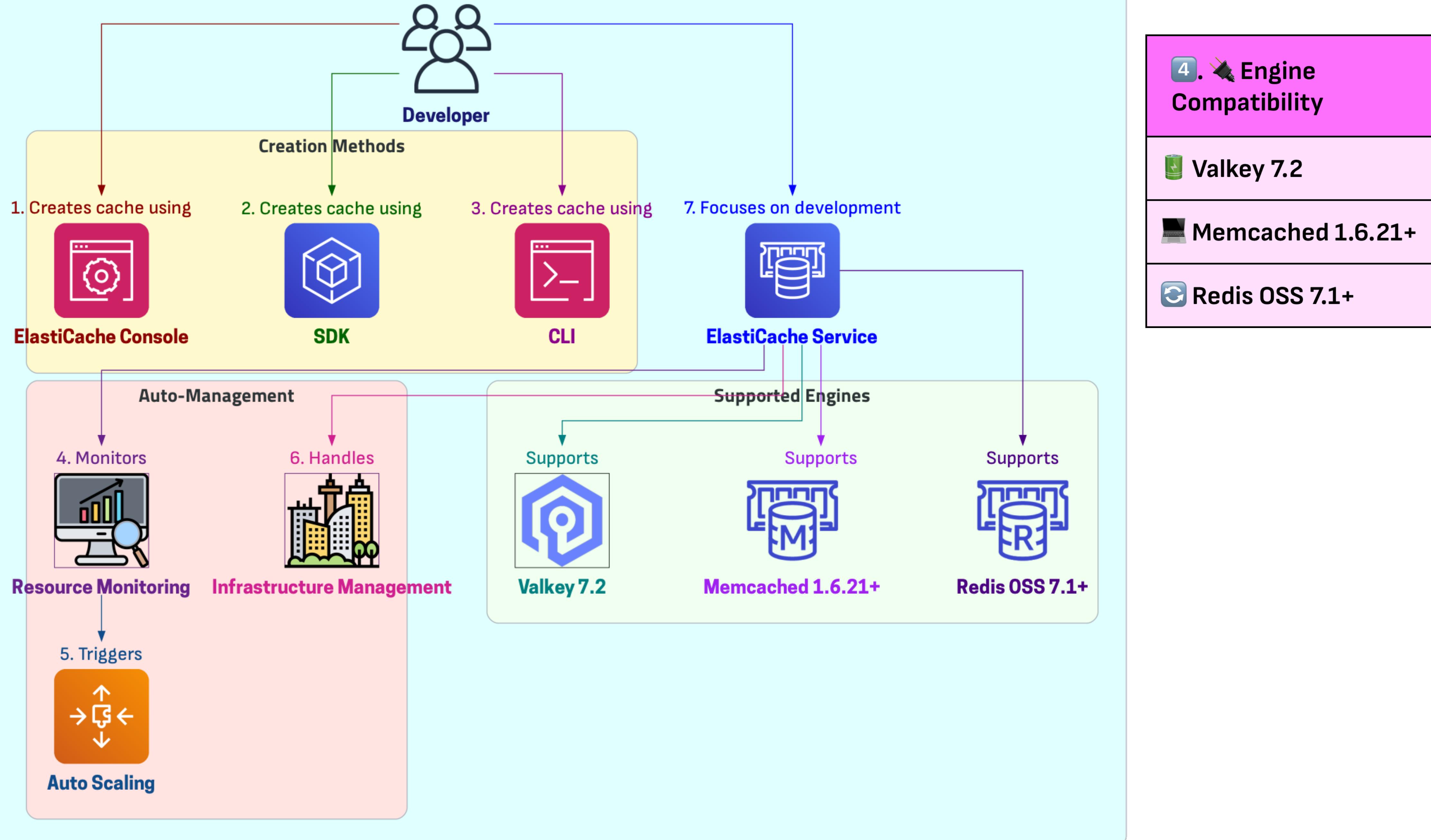
Hardware provisioning

System monitoring

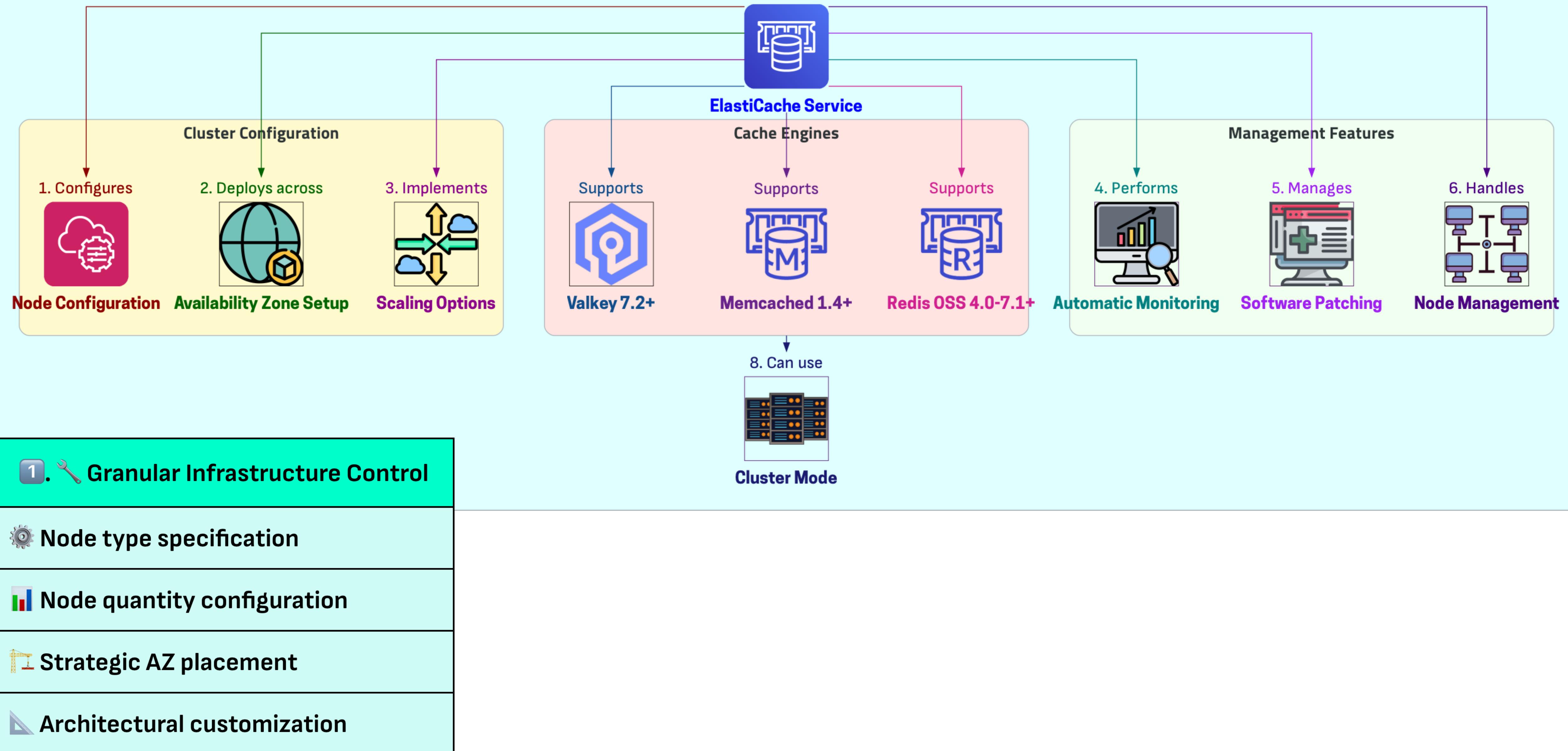
Node replacement

Software patches

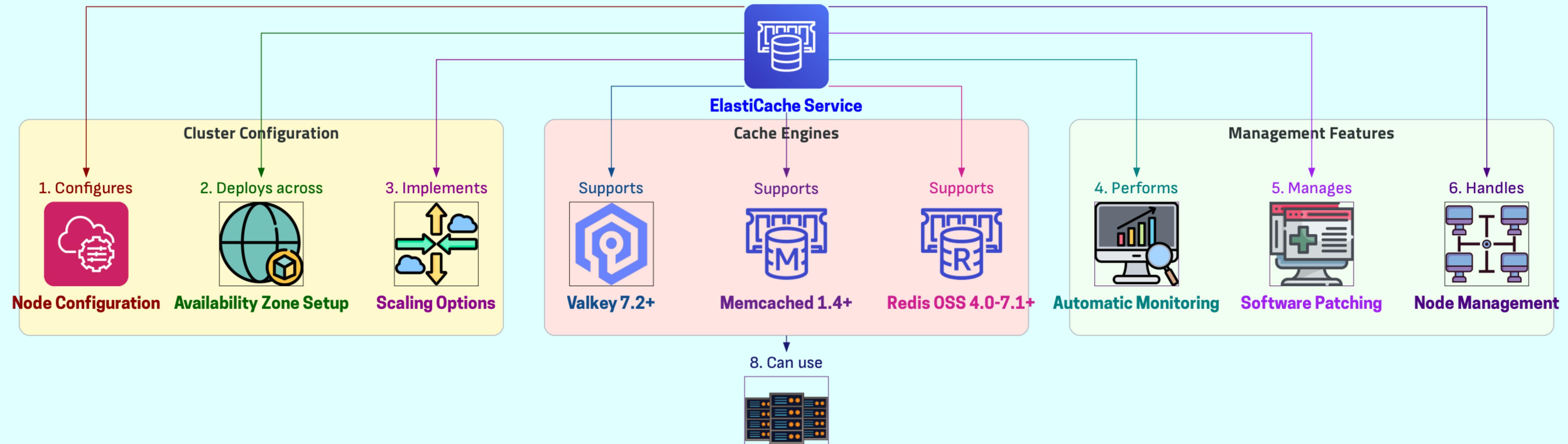
# ElastiCache Serverless



# Designing Custom ElastiCache Clusters



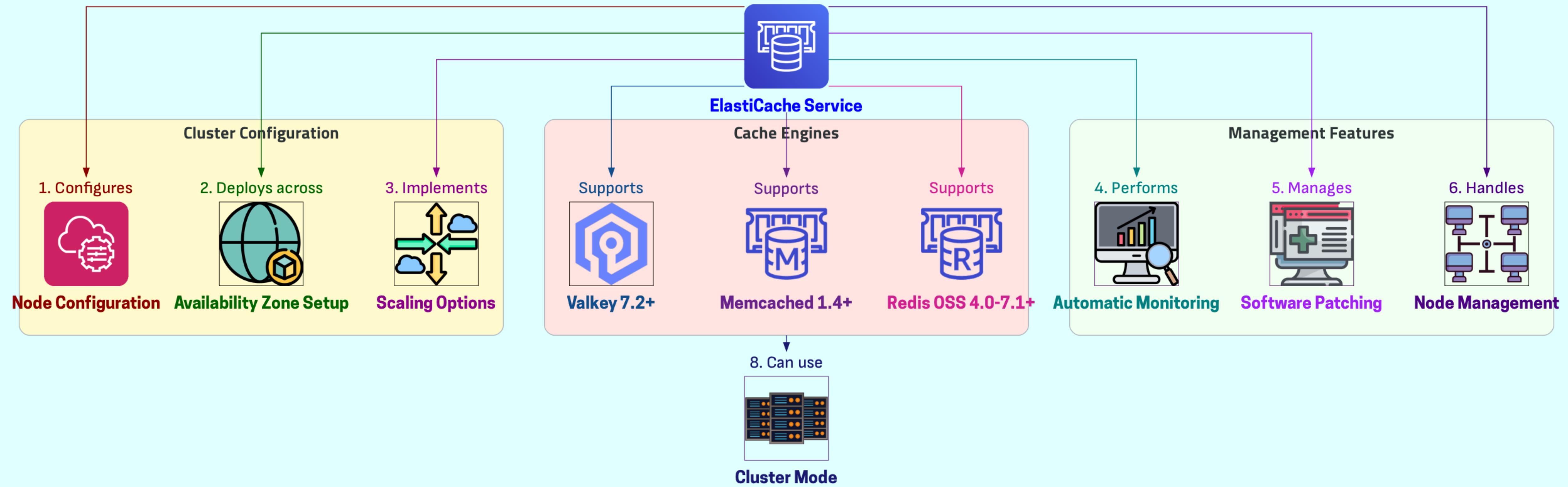
# Designing Custom ElastiCache Clusters



## 2. Flexible Scaling Options

Availability configurations	Single-AZ deployment
	Multi-AZ setup
Scaling modes	Horizontal scaling with cluster mode
	Vertical scaling without cluster mode

# Designing Custom ElastiCache Clusters



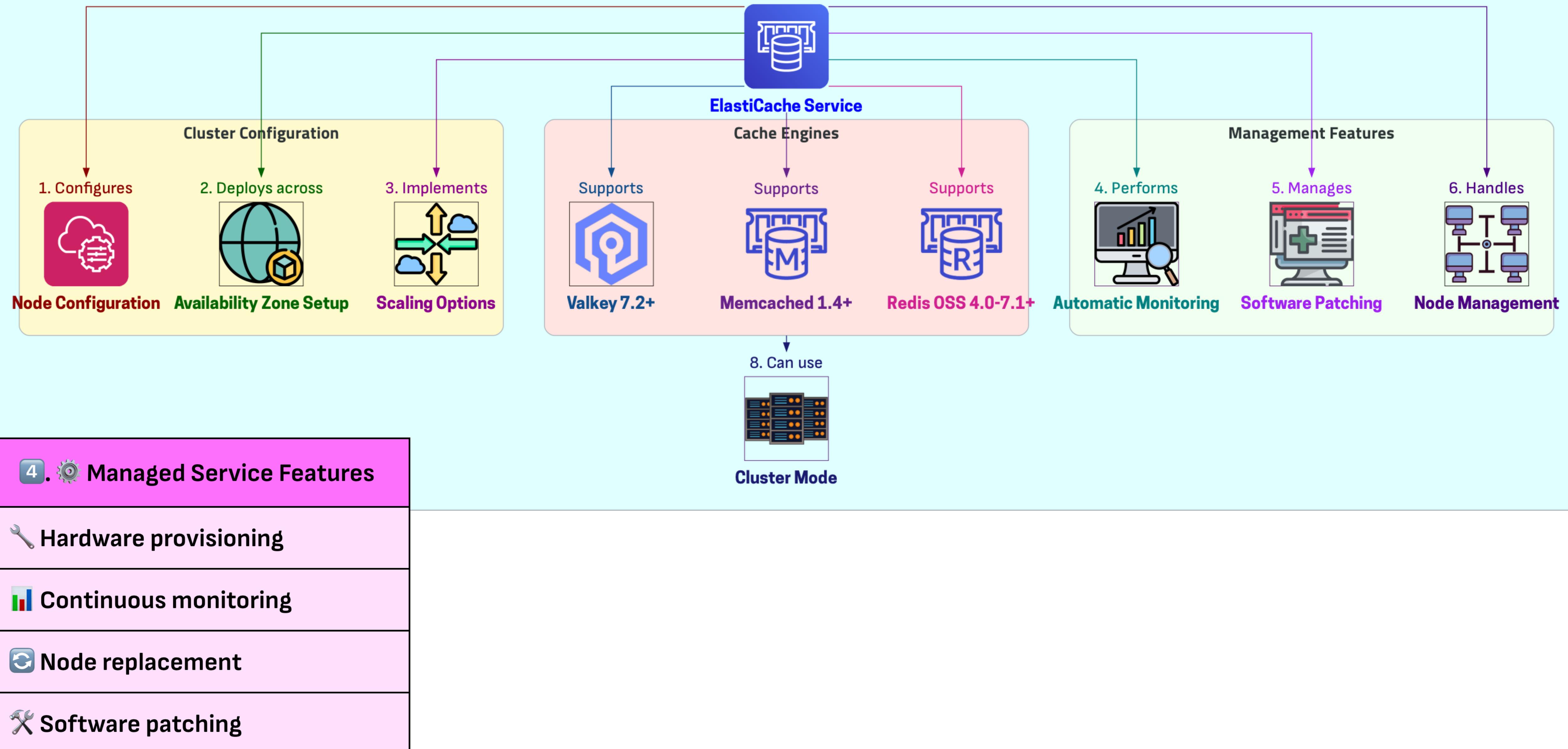
## 3. 🚀 Engine Support

⚡ Valkey 7.2+

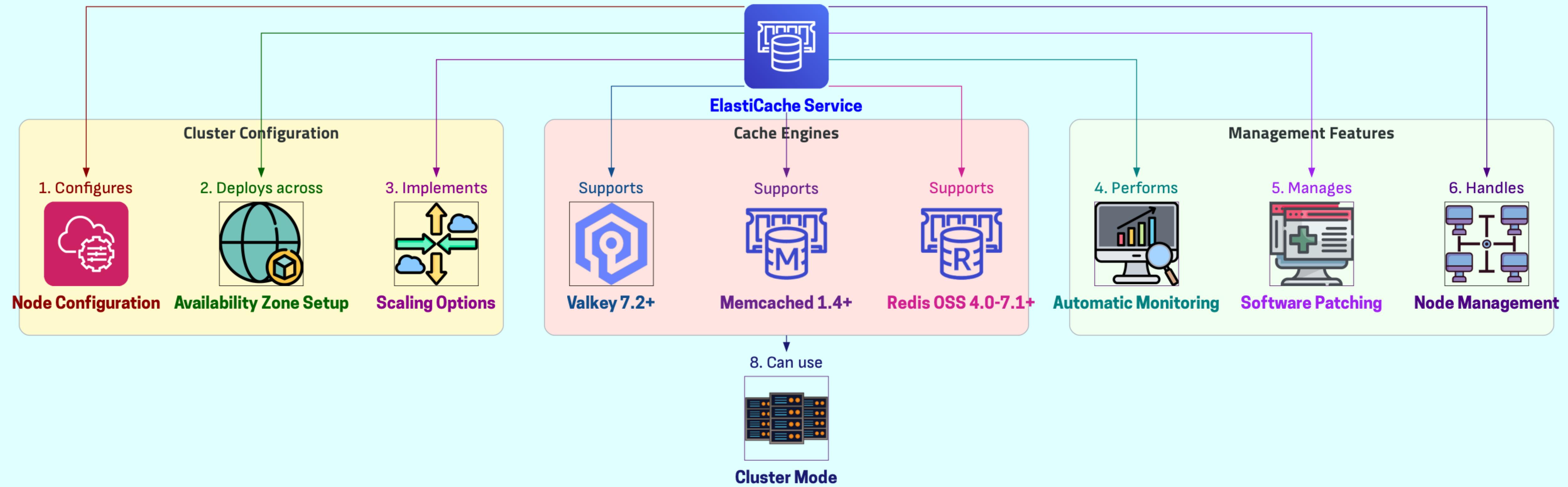
💾 Memcached 1.4+

🔄 Redis OSS 4.0-7.1+

# Designing Custom ElastiCache Clusters



# Designing Custom ElastiCache Clusters



## 5. Capacity Management

Node sizing decisions

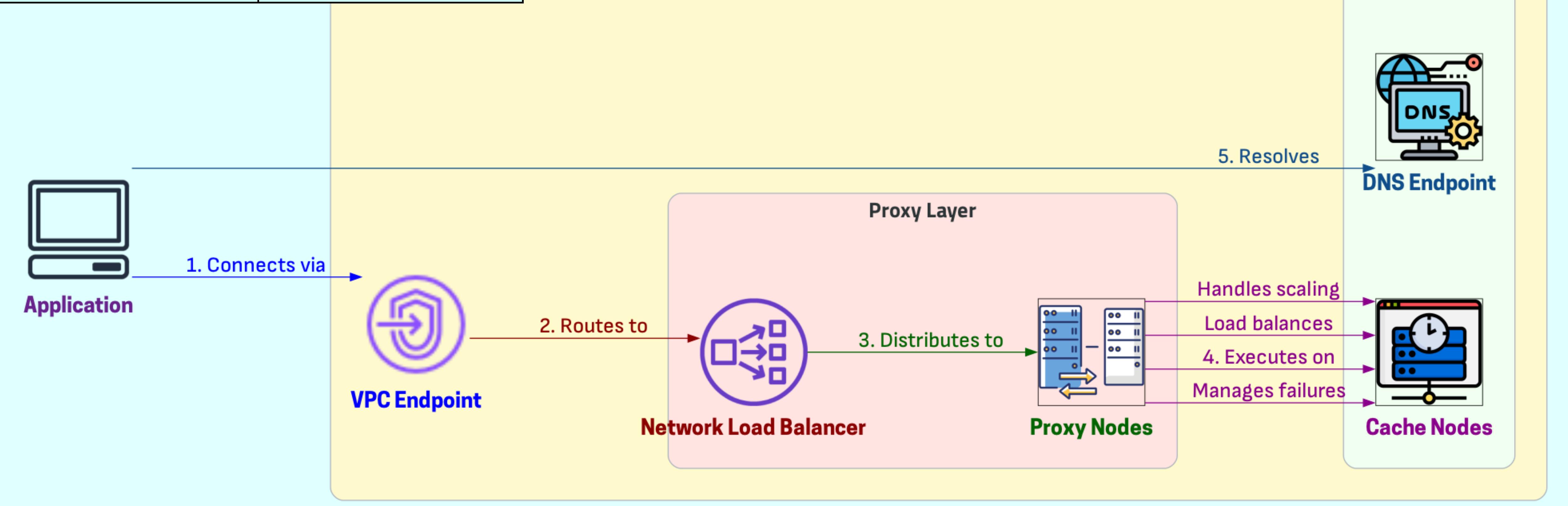
Resource allocation

Patch scheduling

# How ElastiCache Serverless Works

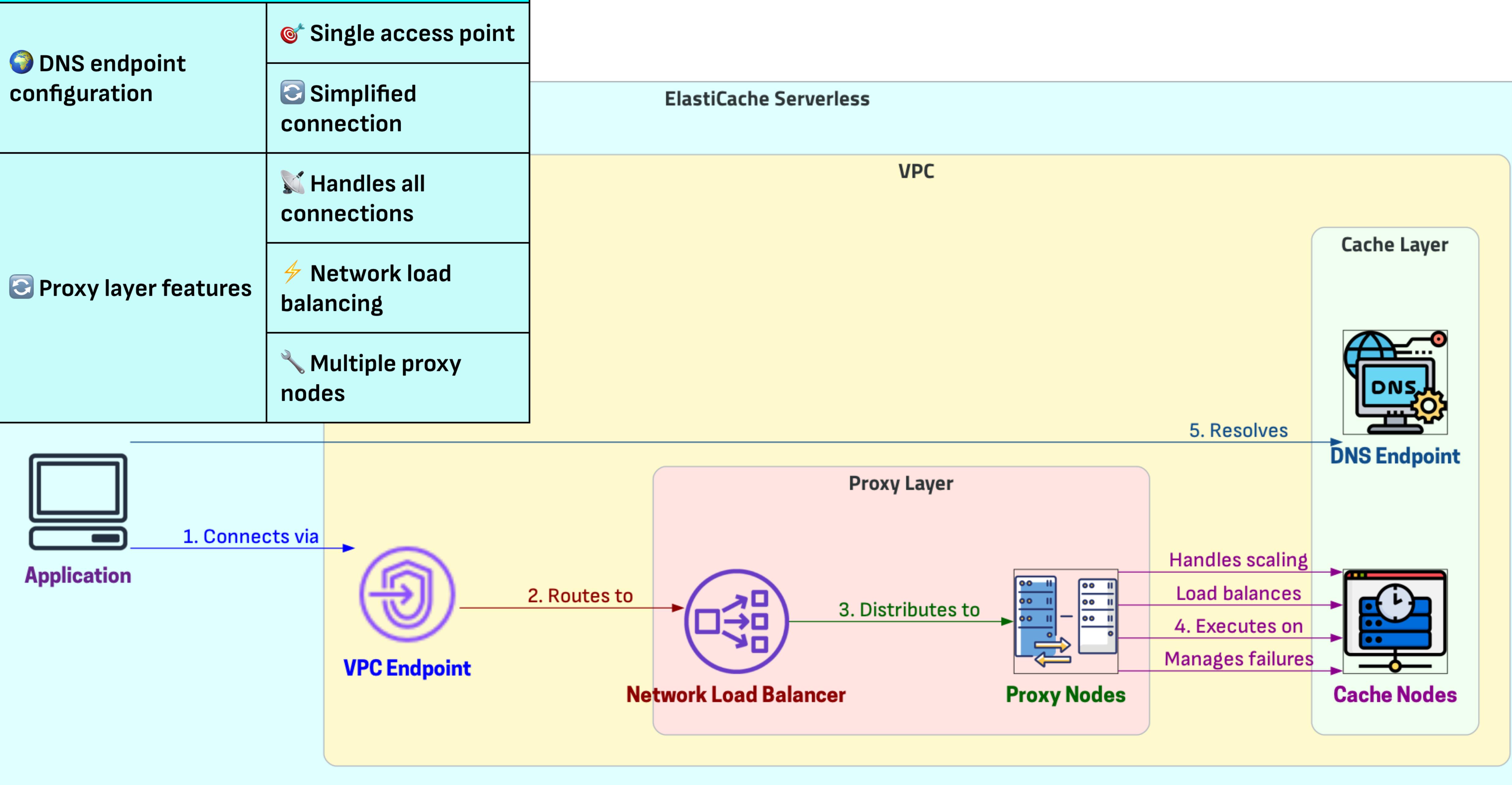
## 1. Network Setup

 VPC infrastructure	 Creates VPC Endpoints  Subnet integration
 Application connectivity	 Secure access  Direct connection



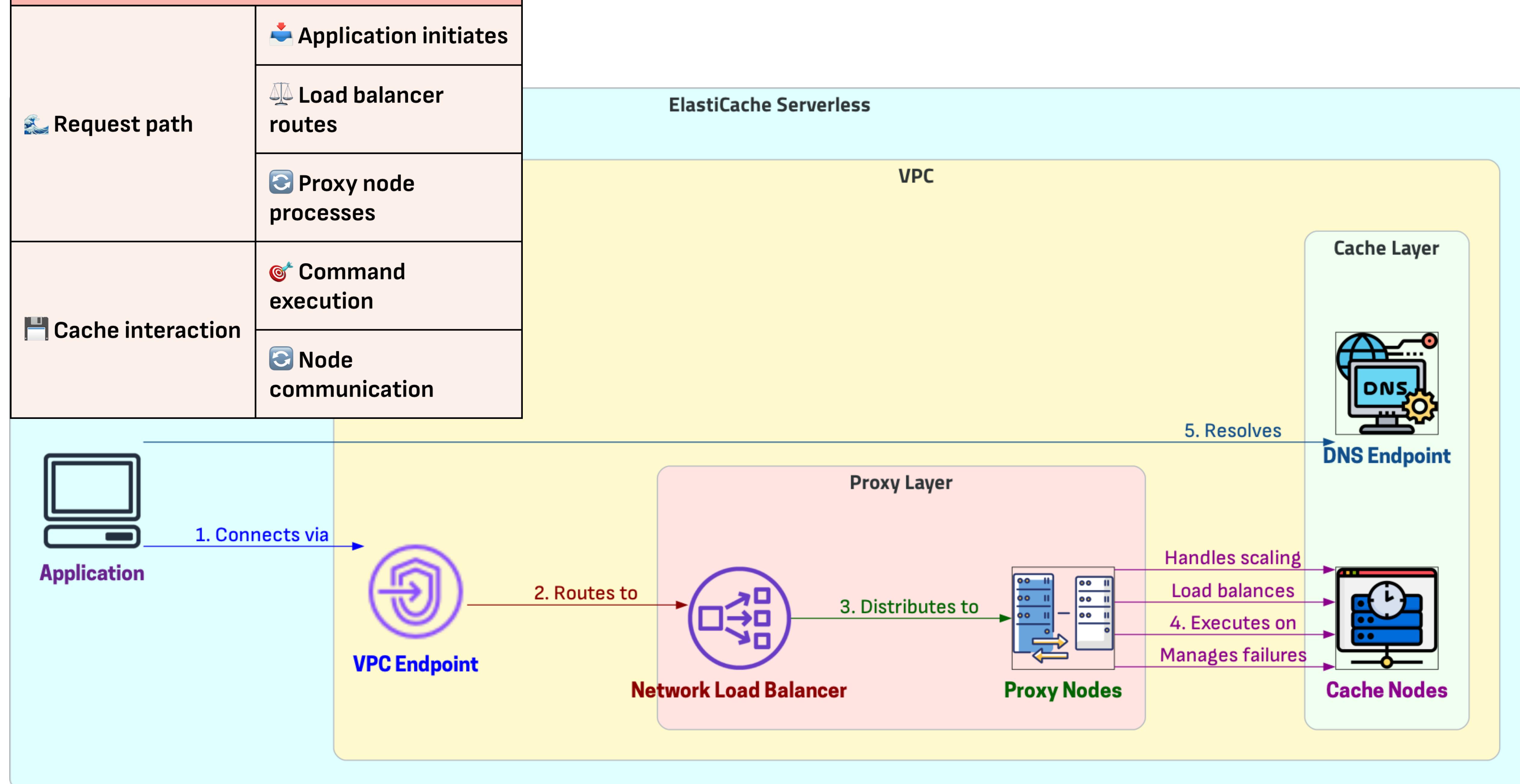
# How ElastiCache Serverless Works

## 2. Single Endpoint Access



### 3. ⚡ Connection Flow

# How ElastiCache Serverless Works



#### 4. 🛡️ Abstraction Layer

🔒 Hidden complexity

🚫 No topology exposure

🎯 Simplified client view

⚖️ Load balancing

📈 Automatic scaling

🔄 Node replacement

🔧 Software updates

# How ElastiCache Serverless Works

## ElastiCache Serverless

VPC

Cache Layer



DNS Endpoint

5. Resolves



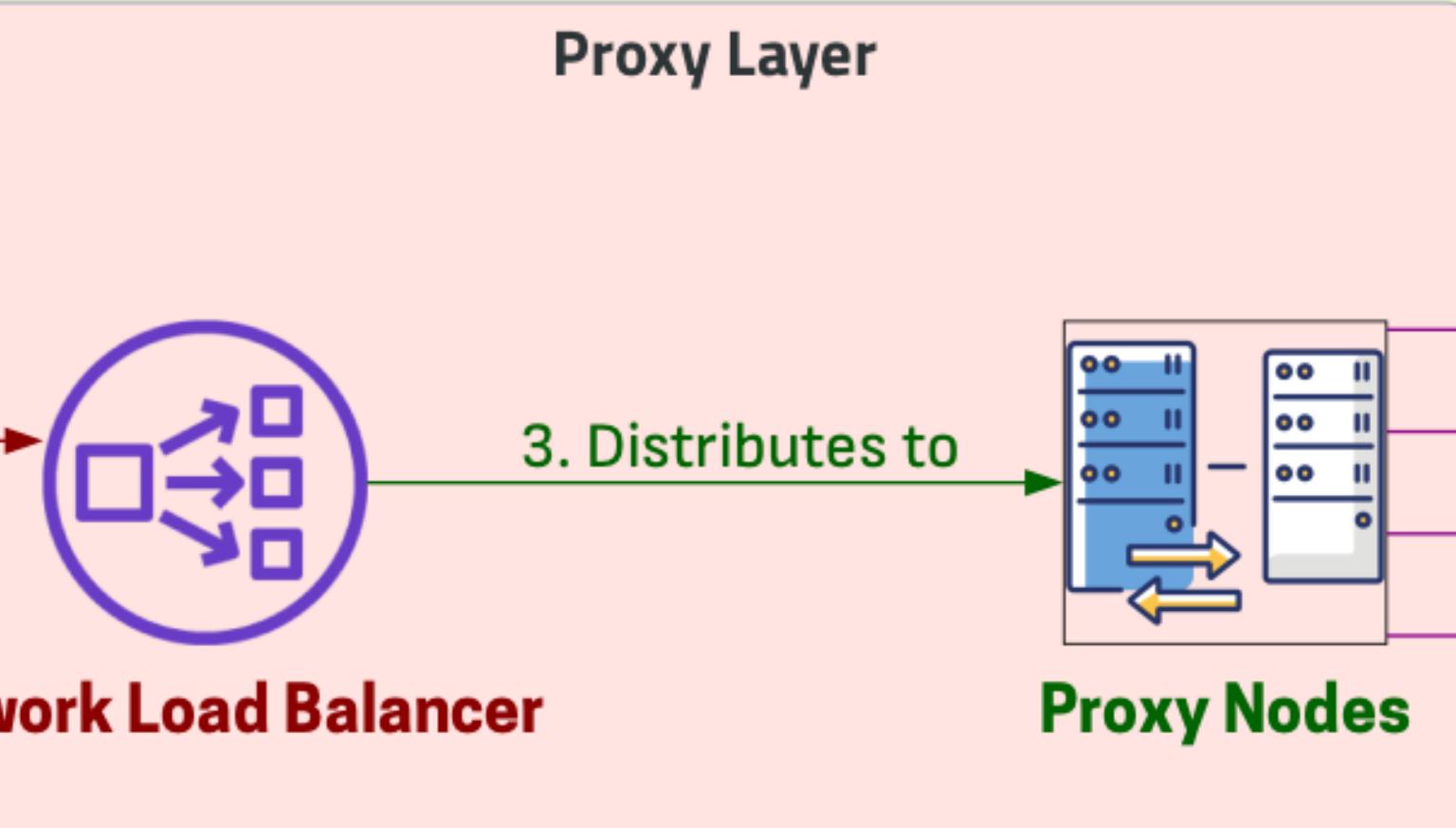
Application

1. Connects via

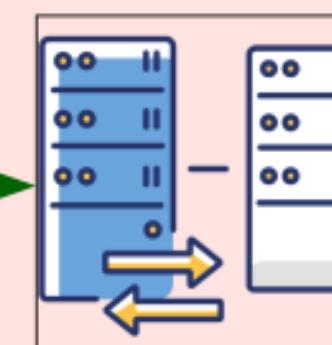


VPC Endpoint

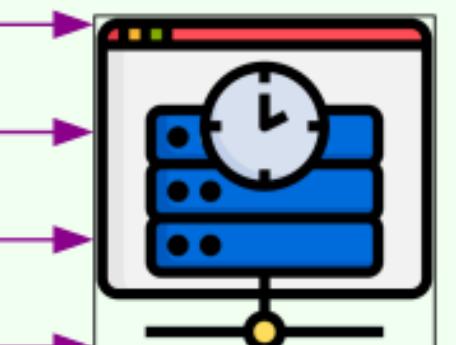
2. Routes to



3. Distributes to

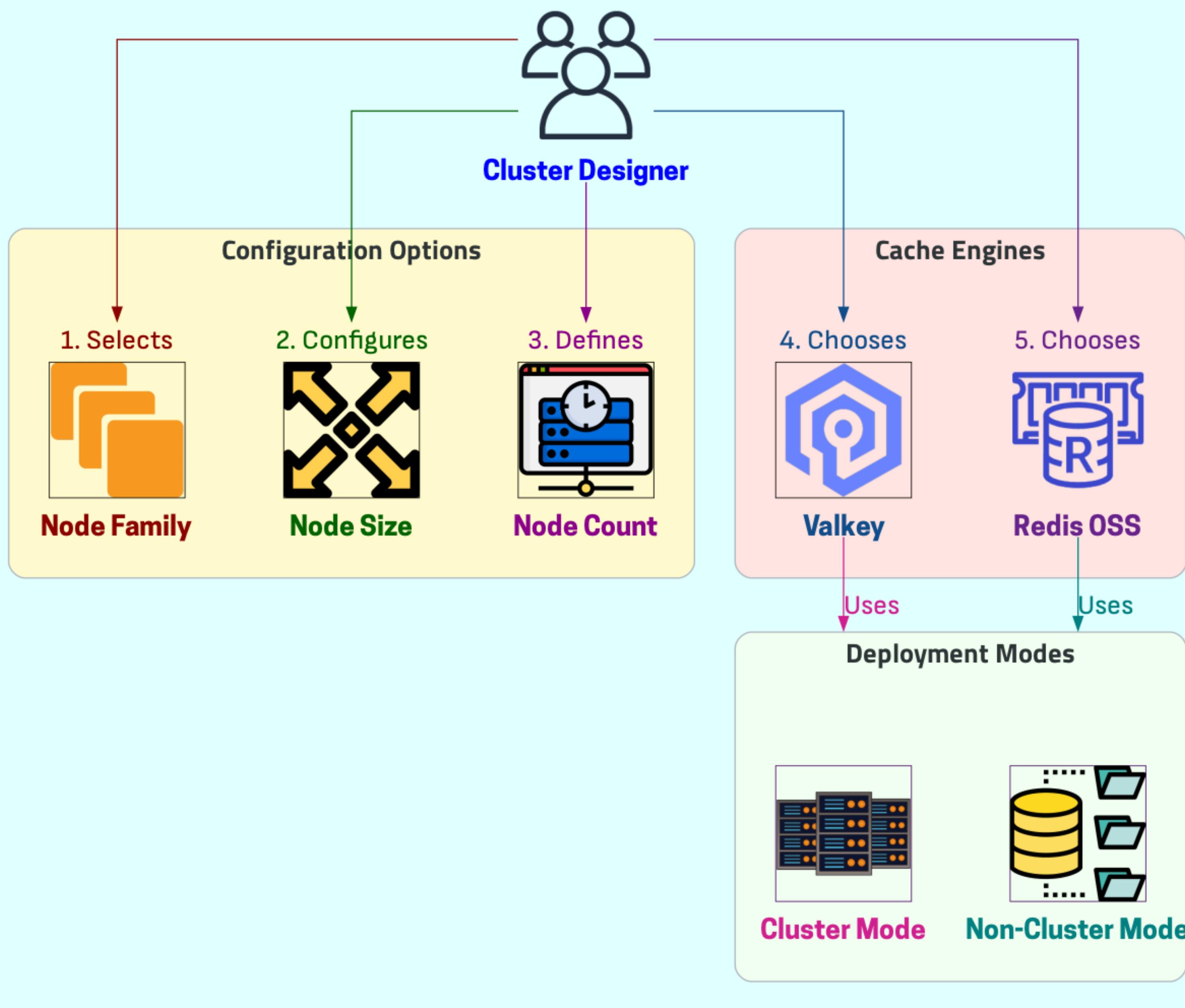


Proxy Nodes



Cache Nodes

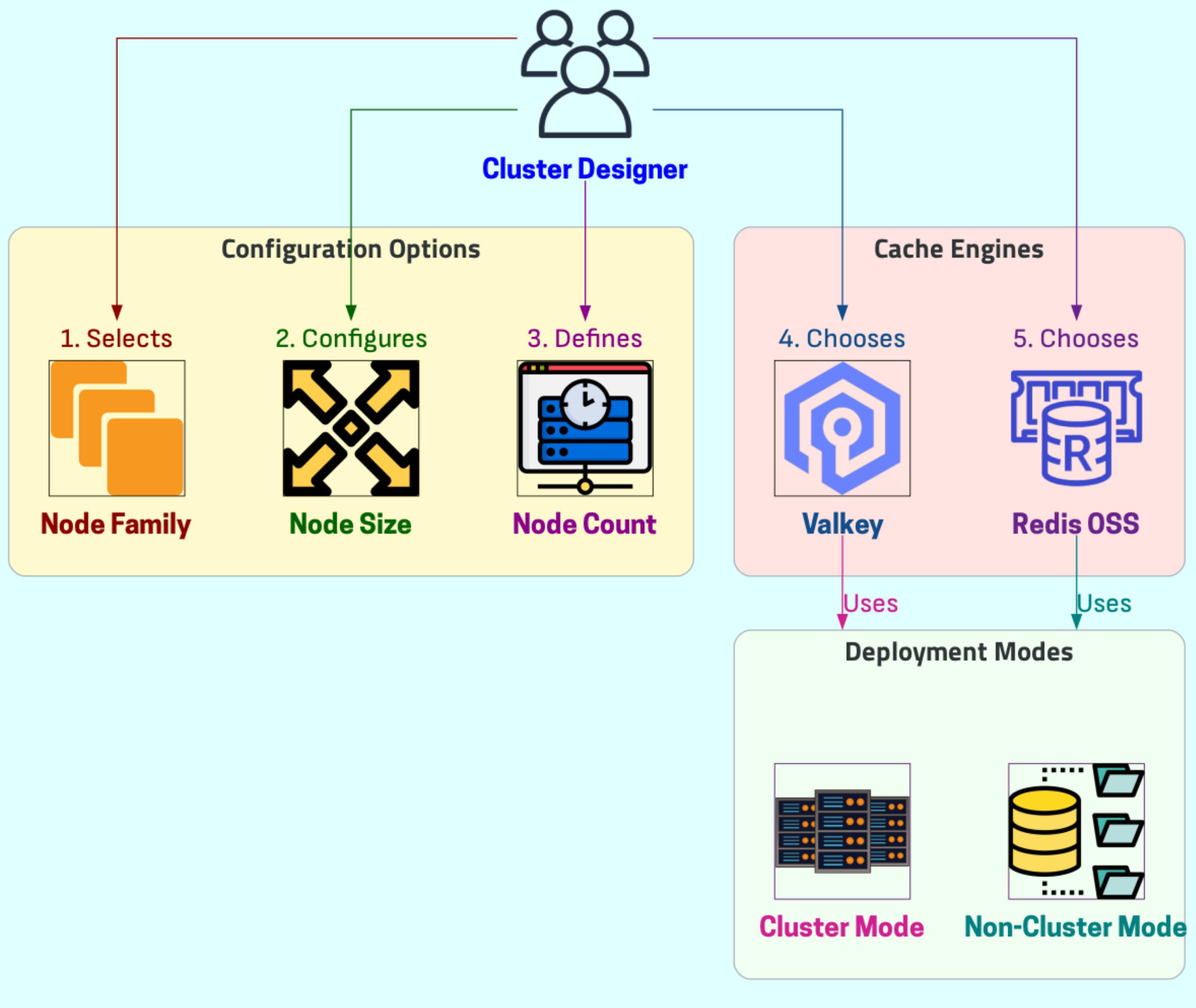
## ElastiCache Self-Designed Cluster



## How ElastiCache self-designed clusters work

1. 🖥️ Node Configuration	
Cache node family selection	⚡ Cache node family selection
Size specifications	📐 Size specifications
Total node count	🔢 Total node count
Core parameters	
Infrastructure optimization	⚡ Infrastructure optimization
Custom tailoring	🎯 Custom tailoring
Precise control	🔧 Precise control

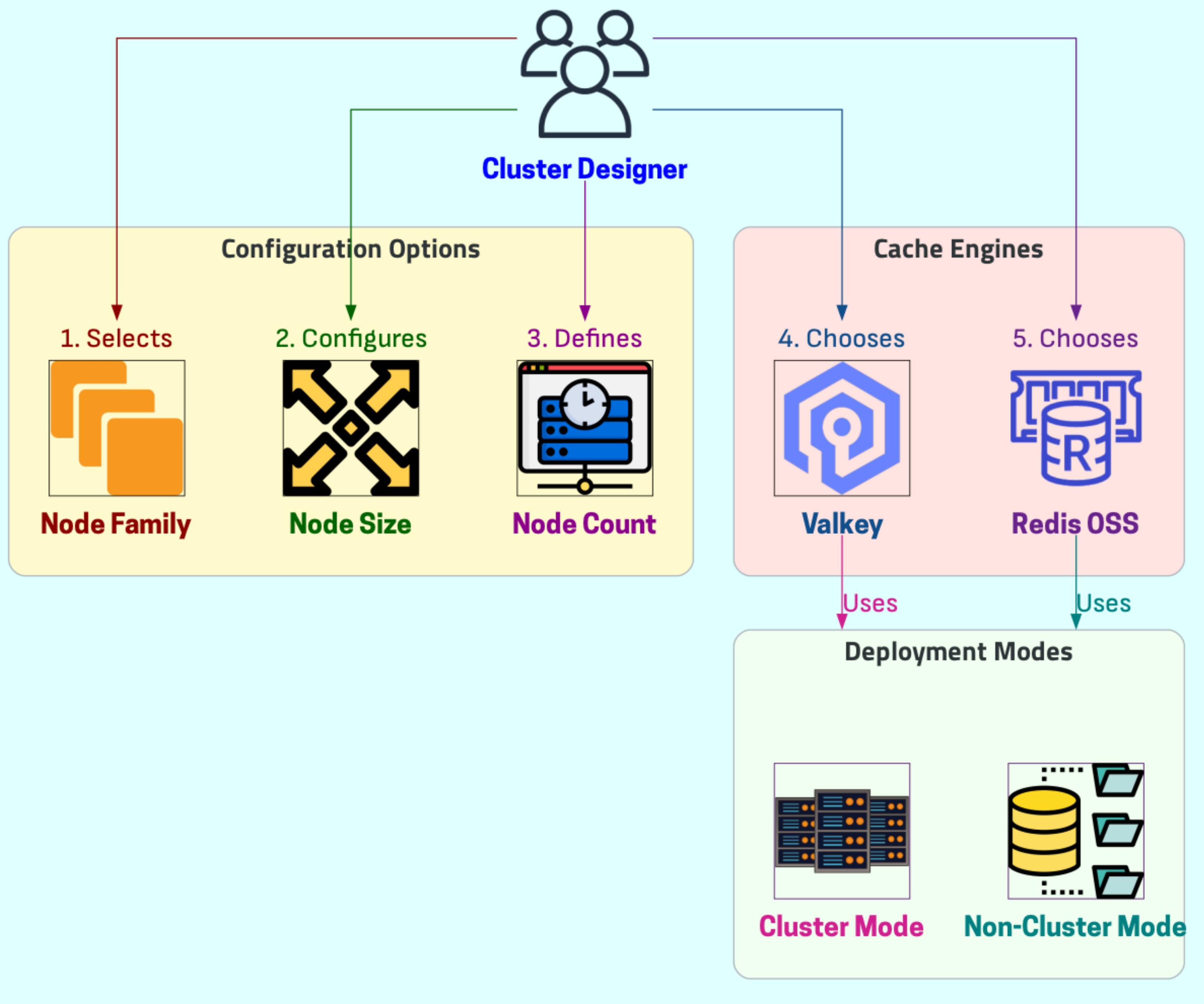
## ElastiCache Self-Designed Cluster



## How ElastiCache self-designed clusters work

2. Shard Management	
<b>Shard configuration</b>	Total shard count
	Node distribution
<b>Node roles</b>	Primary nodes
	Replica nodes

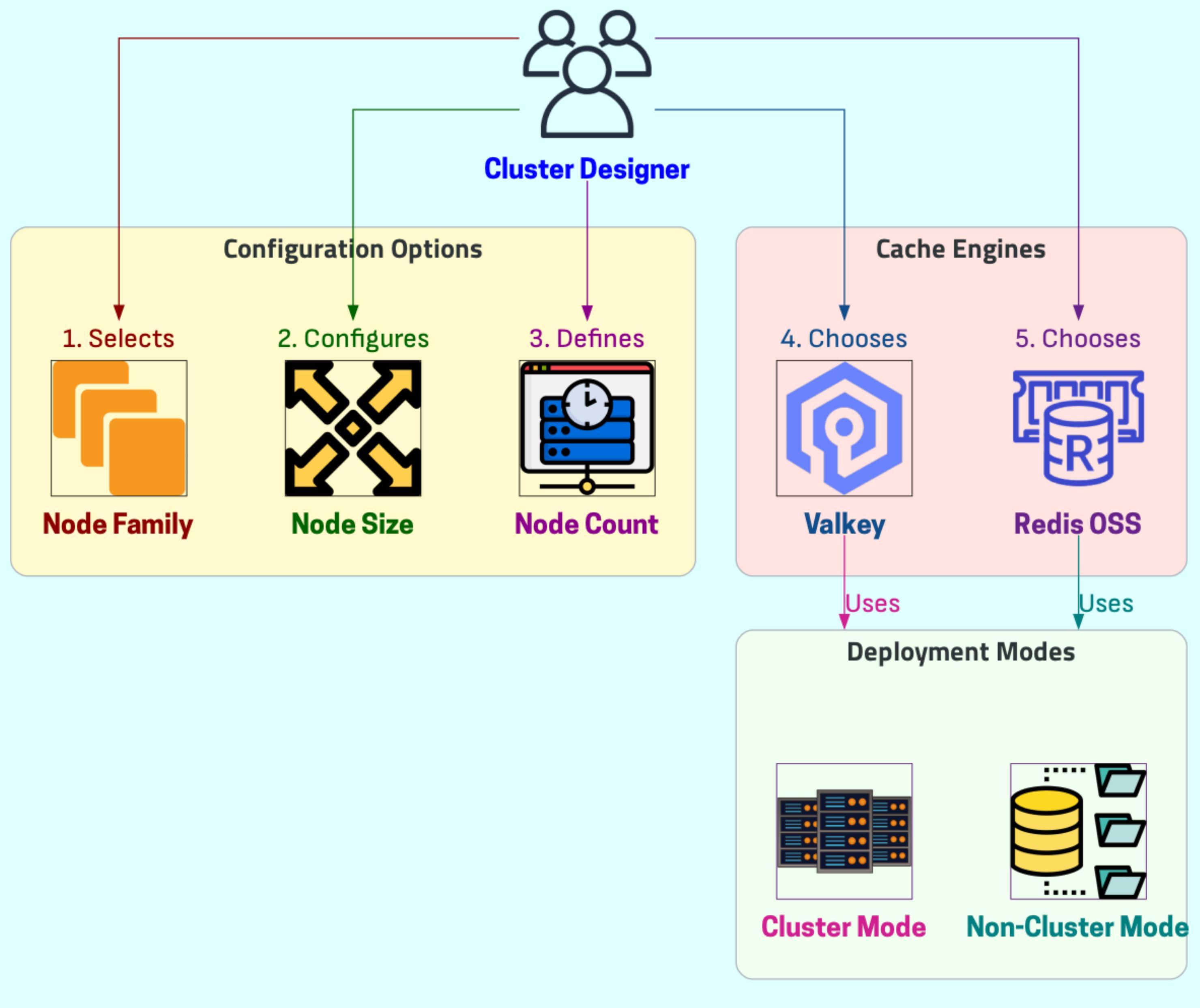
## ElastiCache Self-Designed Cluster



## How ElastiCache self-designed clusters work

3. ⚡ Deployment Options	
<b>Cluster mode</b>	<b>Multiple shards</b>
	<b>Distributed caching</b>
<b>Non-cluster mode</b>	<b>Single shard</b>
	<b>Simpler use cases</b>

## ElastiCache Self-Designed Cluster



## How ElastiCache self-designed clusters work

4. Engine Support	
Supported engines	Valkey compatibility
	Redis OSS support
	Mode-independent
	Consistent options

# Understanding ECPUs Consumption in ElastiCache

## ElastiCache Processing Units Metric

Helps understand ECPUs consumption

## Memcached Commands

Consumes ECPUs based on items and data transfer



## Simple Reads/Writes

Consumes 1 ECPUs per KB transferred

## Valkey/Redis OSS Commands

Consumes ECPUs based on vCPU time or data transfer

### 1. Basic ECPUs Calculation

1 ECPUs per KB transferred

GET: 1KB = 1 ECPUs

SET: 3.2KB = 3.2 ECPUs

### 2. Valkey/Redis Complex Operations

Higher dimension billing

vCPU time consumption

Data transfer volume

3x vCPU + 3.2KB = 3.2 ECPUs

Billing examples

3x vCPU + 2KB = 3 ECPUs

# Understanding ECPUs Consumption in ElastiCache

## ElastiCache Processing Units Metric

Helps understand ECPUs consumption

## Memcached Commands

Consumes ECPUs based on items and data transfer



## Simple Reads/Writes

Consumes 1 ECPUs per KB transferred

## Valkey/Redis OSS Commands

Consumes ECPUs based on vCPU time or data transfer

### 3. ⚡ Valkey/Redis CPU Scaling

Proportional consumption

HMGET example

3x CPU = 3 ECPUs

### 4. 📂 Memcached Multi-Operations

Item-based scaling

1234 3 items = 3 ECPUs

Linear scaling

# Understanding ECPUs Consumption in ElastiCache

## ElastiCache Processing Units Metric

Helps understand ECPUs consumption

## Memcached Commands

Consumes ECPUs based on items and data transfer



## Simple Reads/Writes

Consumes 1 ECPU per KB transferred

## Valley/Redis OSS Commands

Consumes ECPUs based on vCPU time or data transfer

## 5. Memcached Complex Operations

Higher dimension billing

Examples

Item count impact

Data volume impact

3 items + 3.2KB = 3.2 ECPUs

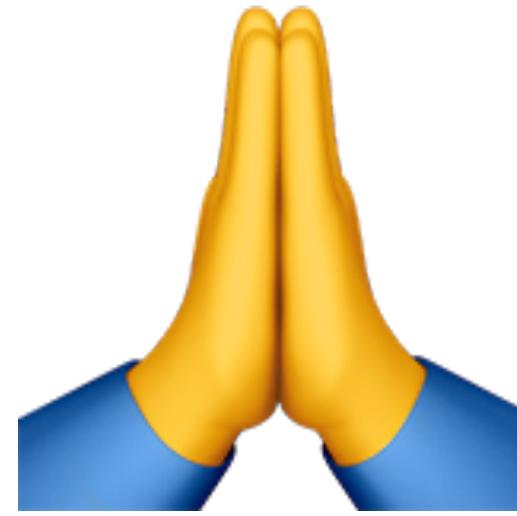
3 items + 2KB = 3 ECPUs

## 6. Monitoring

ElastiCacheProcessing Units metric

Consumption tracking

Workload analysis



**Thanks  
for  
watching**