# Exam: Sample Case Study

During the exam for the Cloud Architect Certification, some of the questions may refer you to a case study that describes a fictitious business and solution concept to provide additional context to exam questions.

- **EHR Healthcare** - https://services.google.com/fh/files/blogs/master_case_study_ehr_healthcare.pdf

  EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi-national medical offices, hospitals, and insurance providers.

  - ❖ Cloud Architecture for on-prem to cloud migration solution & scale problem

- **Helicopter Racing League**- https://services.google.com/fh/files/blogs/master_case_study_helicopter_racing_league.pdf

  Helicopter Racing League (HRL) is a global sports league for competitive helicopter racing. Each year HRL holds the world championship and several regional league competitions where teams compete to earn a spot in the world championship. HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race.

  - ❖ Cloud AI & ML, telemetry and streaming problem

- **TerramEarth** - https://services.google.com/fh/files/blogs/master_case_study_terramearth.pdf

  TerramEarth manufactures heavy equipment for the mining and agricultural industries.. They currently have over 500 dealers and service centers in 100 countries. There are 2 million TerramEarth vehicles in operation currently, and they see 20% yearly growth. Their mission is to build products that make their customers more productive.

  - ❖ Cloud Automation, Operations and API Ecosystem problem

- **Mountkirk Games** - https://services.google.com/fh/files/blogs/master_case_study_mountkirk_games.pdf

  Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They have recently started expanding to other platforms after successfully migrating their on-premises environments to Google Cloud. Mountkirk Games is building a new multiplayer game that they expect to be very popular.

  - ❖ Cloud Auto-scaling, Gaming analytics Problem

# Case Study: EHR Healthcare

## EHR Healthcare Background

**Background**:
- Electronic Health record software to the medical industry
- Provides software as a service to multi-national medical offices, hospitals and insurance providers

**Use Case Statement**:

- EHR Healthcare's business has been growing exponentially year over year
- Need to scale environments, adapt disaster recovery plan, roll out new CD capabilities.
- Google Cloud has been chosen to replace their current colocation facilities.

## Current State

**Environment**:
- Multiple colocation facilities. The lease on one of the data centers is about to expire.

Application:
- Web-based application, many of them run on a group of Kubernetes clusters
- Data is stored in MySQL, MS SQL Server, Redis and MongoDB
- EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. (to be replaced over next several years)
- Monitoring is done via open source tools and email alerts are often ignored.

## Future State Requirements

### Business

**Requirements**:
- On-board new insurance providers as quickly as possible.
- Provide a minimum 99.9% availability for all customer-facing systems.
- Provide centralized visibility and proactive action on system performance and usage.
- Increase ability to provide insights into healthcare trends.
- Reduce latency to all customers.
- Maintain regulatory compliance.
- Decrease infrastructure administration costs.
- Make predictions and generate reports on industry trends based on provider data.

### Technical

**Requirements:**
- Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- Provide a consistent way to manage customer-facing applications that are container-based.
- Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- Provide consistent logging, log retention, monitoring, and alerting capabilities.
- Maintain and manage multiple container-based environments.
- Dynamically scale and provision new environments.
- Create interfaces to ingest and process data from new providers.

## Executive statement

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

# Case Study: Helicopter Racing League

## Helicopter Racing League (HRL) Background

**Background**:
- Global sports league for competitive helicopter racing.
- World championship and several regional league competitions.
- HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race

**Use Case Statement**:
- Migrate existing service to a new platform to expand their use of managed AI & ML
- Deliver serving of the content (real-time & recorded) close to users, particularly in emerging regions.

## Current State

**Solution**:
- Public cloud-first company.
- Video recording and editing is performed at the race tracks, and the content is encoded and transcoded, where needed, in the cloud.
- Enterprise-grade connectivity and local compute is provided by truck-mounted mobile data centers.

**Technical**:
- Existing content is stored in an object storage service on their existing public cloud provider.
- Video encoding and transcoding is performed on VMs created for each job.
- Race predictions are performed using TensorFlow running on VMs in the current public cloud provider.

## Future State Requirements

### Business

HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets. Their requirements are:

- Support ability to expose the predictive models to partners.
- Increase predictive capabilities during and before races:
  - Race results
  - Mechanical failures
  - Crowd sentiment
- Increase telemetry and create additional insights.
- Measure fan engagement with new predictions.
- Enhance global availability and quality of the broadcasts.
- Increase the number of concurrent viewers.
- Minimize operational complexity.
- Ensure compliance with regulations.
- Create a merchandising revenue stream.

### Technical

- Maintain or increase prediction throughput and accuracy.
- Reduce viewer latency.
- Increase transcoding performance.
- Create real-time analytics of viewer consumption patterns and engagement.
- Create a data mart to enable processing of large volumes of race data.

## Executive statement

Our CEO, S. Hawke, wants to bring high-adrenaline racing to fans all around the world. We listen to our fans, and they want enhanced video streams that include predictions of events within the race (e.g., overtaking). Our current platform allows us to predict race outcomes but lacks the facility to support real-time predictions during races and the capacity to process season-long results.

# Case Study: TerramEarth

## TerramEarth Background

**Background**:
- Mining and Agricultural equipment Manufacturer
- 500 dealers and service centers in 100 countries

**Use Case Statement**:
- 2 million vehicles and projected 20% yearly growth
- Vehicles collect Telemetry data from many sensors during operation.
- Small subset of critical data transmitted in real time from vehicles to felicitate fleet management.
- Rest of the sensor data is collected, compressed and uploaded daily at home base.
- 200-500 MB of data generated by each vehicle every day

## Current State

**Existing Technical environment**:

- TerramEarth's vehicle data aggregation and analysis infrastructure resides in Google Cloud and serves clients from all around the world.
- A growing amount of sensor data is captured from their two main manufacturing plants and sent to private data centers that contain their legacy inventory and logistics management systems.
- The private data centers have multiple network interconnects configured to Google Cloud.
- The web frontend for dealers and customers is running in Google Cloud and allows access to stock management and analytics.

## Future State Requirements

### Business

**Requirements**:

- Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-intime repair where possible.
- Decrease cloud operational costs and adapt to seasonality.
- Increase speed and reliability of development workflow.
- Allow remote developers to be productive without compromising code or data security.
- Create a flexible and scalable platform for developers to create custom API services for dealers and partners.

### Technical

**Requirements**:

- Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.
- Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.
- Allow developers to run experiments without compromising security and governance requirements
- Create a self-service portal for internal and partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- Use cloud-native solutions for keys and secrets management and optimize for identity based access.
- Improve and standardize tools necessary for application and network monitoring and troubleshooting

## Executive statement

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize vehicle downtimes. After moving multiple systems into Google Cloud, we are seeking new ways to provide best-in-class online fleet management services to our customers and improve operations of our dealerships. Our 5-year strategic plan is to create a partner ecosystem of new products by enabling access to our data, increasing autonomous operation capabilities of our vehicles, and creating a path to move the remaining legacy systems to the cloud.

# Case Study: Mountkirk Games

## MountKirk Games Background

**Background**:

- Online, session-based, multiplayer games maker for mobile platforms
- Started expanding to other platforms after successfully migrating their on-premises environments to Google Cloud
- Recent Endeavor is to create a retro-style first-person shooter(FPS) game that allows hundreds of simultaneous players to join a geo-specific digital arena from multiple platforms and locations. A real-time digital banner will display a global leader board of all the top players across every active arena.

**Use Case Statement**:

- Building a new multiplayer game which is expected to be very popular
- Deploy game's backend on Google Kubernetes Engine to scale rapidly and use Google's global load balancer to route players to the closest regional game arenas.
- In order to keep the global leader board in sync, they plan to use a multi-region Spanner cluster

## Current State

**Existing technical environment**:

- Recently migrated to Google Cloud.
- Five games came across using lift-and-shift virtual machine migrations, with a few minor exceptions.
- Each new game exists in an isolated Google Cloud project nested below a folder that maintains most of the permissions and network policies.
- Legacy games with low traffic have been consolidated into a single project.
- There are also separate environments for development and testing

## Future State Requirements

### Business

**Requirements**:

- Support multiple gaming platforms.
- Support multiple regions.
- Support rapid iteration of game features.
- Minimize latency.
- Optimize for dynamic scaling.
- Use managed services and pooled resources.
- Minimize costs.

### Technical

**Requirements**:

- Dynamically scale based on game activity.
- Publish scoring data on a near real–time global leaderboard.
- Store game activity logs in structured files for future analysis.
- Use GPU processing to render graphics server-side for multi-platform support.
- Support eventual migration of legacy games to this new platform.

## Executive statement

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles. Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.