**Overview.**

It's your first day at Cie and you're tasked with constructing a high-level architecture with accompanying justification and documentation for a future SaaS platform. Be prepared to support your decisions during a follow-up Interview (either on site or virtual) with Cie's technical and project teams.

**Problem Statement.**

**ReadyPitchGo** Is a (**fictitious**) Cie startup that Is tasked with creating a SaaS platform that will need to support every major league, minor league, collegiate and high school baseball teams. There are an estimated 40,000 teams each with 15 pitchers per year that will need to be monitored. This platform's main purpose Is to track the performance of a team's pitchers by tracking their velocity, the type of pitch and the effectiveness.

*The* ***ReadyPitchGo*** *team will have at least a few developers working on this platform at the same time, so please be cognizant of that when answering some of the questions defined below.*

At the core of **ReadyPitchGo** are the following properties:

* A hardware component (pitch speed and type detection) that will log every pitch from every pitcher (type of pitch, speed of the pitch, whether the pitch was a ball or a strike) for each supported team
  + ***Assume that the hardware component Is configured to hit a specific set of endpoints with configured team Information, pitcher and pitch Information***
* An API or set of APIs for the hardware, reporting and various other properties to Interface with
* A reporting and team admin tool for the teams to login and view high level reports for:
  + Their team: average velocity, pitch types, risky pitchers and other misc. self-service reports
  + Metrics for each pitcher
  + Modify their teams (add/edit/deactivate pitchers)
  + Manage billing
* A **ReadyPitchGo** admin tool to setup teams and view diagnostic Information
* Scheduled tasks for billing and any reporting purposes
* Underlying data storage Including caching, reporting, etc.
* Any other properties determined by the technical team

On the technical roadmap, there are plans for mobile applications that will also Interface with the API/APIs, however their functionality set and release date Is TBD.

With the above Information, you are tasked with defining the following:

* What is the technology stack?
  + What programming languages would you use for the API/APIs? Why?
  + Which database technology(ies) would you leverage? Why?
  + What other platforms would you leverage, if at all? Why?
* Caching strategy
  + What data can/should be cached?
  + What caching frameworks/platforms/approaches would you leverage and In what circumstances?
* API documentation
  + How will you provide documentation for API consumers?
* Exception/logging
  + What frameworks/tooling will you leverage for exception logging? Standard logging?
* Validation
  + What libraries (If any) will you use for validation of data?
  + How would you integrate the validation of user input throughout the various properties to ensure data integrity?
* Authorization and authentication
  + How will you handle authorization and authentication?
  + How will you differentiate between the various types of roles?
  + How would you manage access and security on the API level to restrict access to other team data?
* Security/encryption
  + What types of hashing and/or encryption would you leverage?
  + What data would you hash/encrypt?
  + Do you see any potential problematic areas specifically around areas like data protection, compliancy, etc.?
* Reporting
  + What platforms/tools/approaches will you leverage to support ad-hoc, self-service reports for both teams and administrators?
* Scheduled tasks
  + What platforms/tools/approaches will you use to support scheduled tasks?
  + What types of tasks do you envision needing to be scheduled versus real time?
* Load/Performance testing
  + How will you be confident that the **ReadyPitchGo** platforms can handle the load from 40,000 teams, 600,000 pitchers with hundreds of millions of logged data points?
  + What Is your load and performance testing plan?
* Unit/Integration testing
  + What tools and approaches would you use to ensure that the code Is not just unit testable, but also that tests are ran often?
  + Give a few examples of how you see the difference of unit tests for the **ReadyPitchGo** platform versus integration tests.
* Branching strategy
  + With so many features/functionality being developed concurrently, what branching strategy will you leverage?
* Deployments
  + What Is your plan for managing builds and deployments across the various environments? Which and how many environments would you plan to support?

**Deliverables.**

You're responsible for providing the following:

* Answers to the questions defined under the **Problem Statement** section above.
* Infrastructure diagram(s)
* High level architecture diagram(s)
* Hosting
  + Where it will be hosted (AWS, Azure, Google Cloud, etc.)
  + What services for the cloud platforms that will be utilized and reasoning why
  + Reasoning for selecting that hosting option versus their competitors
* Estimated Hosting Costs
  + Rough hosting costs based on the various environments, planned usage, etc.
* Assumptions
  + Any assumptions that you're making for any question answered
* Outstanding Questions
  + What did Cie miss by putting together the Problem Statement?
* Potential Risks
  + Defining any risks whether they be technical or not
* Technical Roadmap
  + Technical items that you would recommend for future phases of the **ReadyPitchGo** SaaS platform and supporting reasons why.

Your deliverables may be submitted in any format that you'd like just as long as the content (and this test) is kept confidential. For reference, we've taken tests by way of GitHub repositories using Markdown, Microsoft Word, Google Docs and other mediums.

**Important Notes.**

Even though the above test doesn't Involve an actual coding exercise, if you'd like to provide any code snippets or repositories that you feel best represent your overall ability and/or would like to provide relevant coding based on the **ReadyPitchGo** exercise, please submit your code by way of repository links, GitHub Gists or similar mechanisms.

Overall, Cie will be looking for the completeness of your answers and the justification. Please be careful when copying and pasting from other sources as you'll need to be comfortable discussing and explaining the rationale for your answers.

