AESTHETIC RECORD

Technology Choices Q&A

Redis

- What is Redis?
  - In memory data store, best suited for managing sessions and user cache in a structured data fashion.
- Why Redis?
  - Most mature technology for session and cache management.

Elasticsearch

- What is Elasticsearch?
  - Analytics engine best suited for aggregation of time series data (application logs).
- Why Elasticsearch?
  - Most mature technology for aggregating time series data. Easy integration with Splunk.

Splunk

- What is Splunk?
  - Operations dashboard for data visualization.
- Why Splunk?
  - Quickest approach to provide log visualization.

PostgreSQL

- What is PostgreSQL?
  - Relational database storage.
- Why PostgreSQL?
  - Most advanced technology for RDB. Open source technology communities tend to have better documentation, plugins and libraries. Well suited for definitively structured, non-fluid, non-dynamic data (i.e. Billing data, Transactional data, permanently scoped data models).

MongoDB

- What is MongoDB?
  - Document oriented database storage.
- Why MongoDB?
  - Most advanced technology for Document Format storage. Most performant technology when paired with GraphQL application design. Best technology for storage of fluid and dynamic data sources (i.e. User entity, Business entity, Location entity, et al).

ReactJS

- What is ReactJS?

- Contemporary front-end library predicated on virtual DOM manipulation and component-based design.

- Why ReactJS?
  - Legacy; also, preference.

NodeJS

- What is NodeJS?
  - Back-end, JavaScript based run time environment.
- Why NodeJS?
  - Asynchronous processing, with good design, provides improved performance with fetch request handling. KISS design methodology -> single programming language requirement for application. Highly performant benchmarks

Mocha

- What is Mocha?
  - Testing framework for NodeJS and JavaScript based applications.
- Why Mocha?
  - Most mature library for testing JavaScript based applications. Relatively simplistic integration with Jenkins CI/CD.

Chai

- What is Chai
  - An assertion library which complements Mocha and Jenkins.
- Why Chai?
  - Most mature assertion library; makes testing easier to read.

Core Language: JavaScript; Superset Language: TypeScript

- What is JavaScript?
  - Scripting language with standardization practices set by Ecma International.
- What is TypeScript?
  - A JavaScript superset developed by Microsoft that provides the option for static typing.
- Why TypeScript?
  - Type safety on compile; easier to read and understand code. Free type checking if written with Visual Studio Code IDE.

Kafka

- What is Kafka?
  - Stream processing software; also referred to as a Message Bus.
- Why Kafka?
  - Complements asynchronous design well. Allows transformative data processing without the need to keep responsive connections open – assists in reducing user experience of latency.

GraphQL

- What is GraphQL?

  - Data query technology.
- Why GraphQL?
  - Permanently resolves overfetch and underfetch issues. Substantially more performant than REST or SOAP designs. Will reduce latency, provide better benchmarks.

Apollo Client

- What is Apollo Client?
  - A comprehensive state management library that allows management of both local and remote data when paired with GraphQL.
- Why Apollo Client?
  - Most mature library for applications needing state management that use GraphQL.

Hosting: AWS (Amazon Web Services)

- What is AWS?
  - The crème de la crème of web hosting services.
- Why AWS?
  - Most mature and advanced hosting provider on the market.

AWS S3: Directory Storage for On Demand Files

- What is S3?
  - Simple Storage Service; Amazon's directory storage service.
- Why S3?
  - Most mature and advanced directory storage service on the market.

Docker

- What is Docker?
  - Containerization services.
- Why Docker?
  - Containerization ensures the application performs consistently across all platforms. Provides accelerated development, portability, complements microservice architecture. More performant than VM centric hosting.

Kubernetes

- What is Kubernetes?
  - Scalable container orchestration system.
- Why Kubernetes?
  - Provides load balancing, network traffic distribution, storage service management, automation of rollout/rollbacks, direct control of CPU and RAM allotted per container, self-correcting – automates failing container management, vault services – can manage secrets, credentials, tokens and SSH keys.

Git

- What is Git?
  - Source Version Control for application.

- ❖ Why Git?
  - ➢ Most mature technology for source version control. Superior to Mercurial, Subversion, MSFT, among other alternatives.

GitLab (same to GitHub)

- ❖ What is GitLab?
  - ➢ Repository hosting services for Git based applications.
- ❖ Why GitLab?
  - ➢ Most mature service for source version control hosting. Simplistic Jenkins integration, among many other service integration automations. Robust support for repository dependent hooks (i.e. for Jira).

Jenkins CI/CD

- ❖ What is Jenkins CI/CD?
  - ➢ Automation Server for CI/CD (Continuous Integration / Continuous Delivery) service.
- ❖ Why Jenkins?
  - ➢ Most mature service for CI/CD and deployment automation. Expansive library of plugins. Easy integration with other services. Simplistic to use and configure.

AWS S3 Glacier: Long Term Storage for Stale Data

- ❖ What is S3 Glacier?
  - ➢ Simple Storage Service for stale data.
- ❖ Why S3 Glacier?
  - ➢ Most mature service for long term, stale data that is not anticipated to be accessed under any existing business cases.