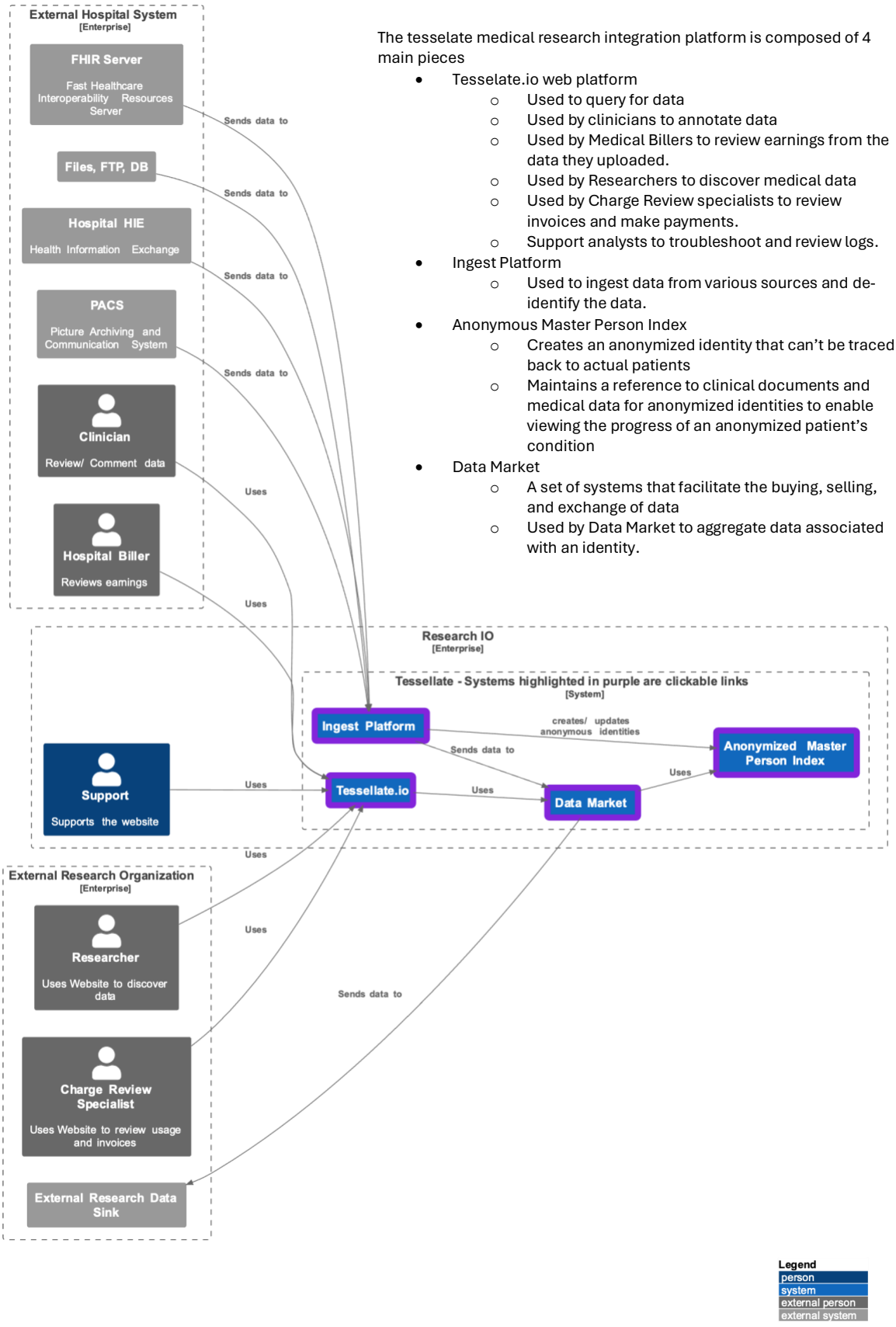
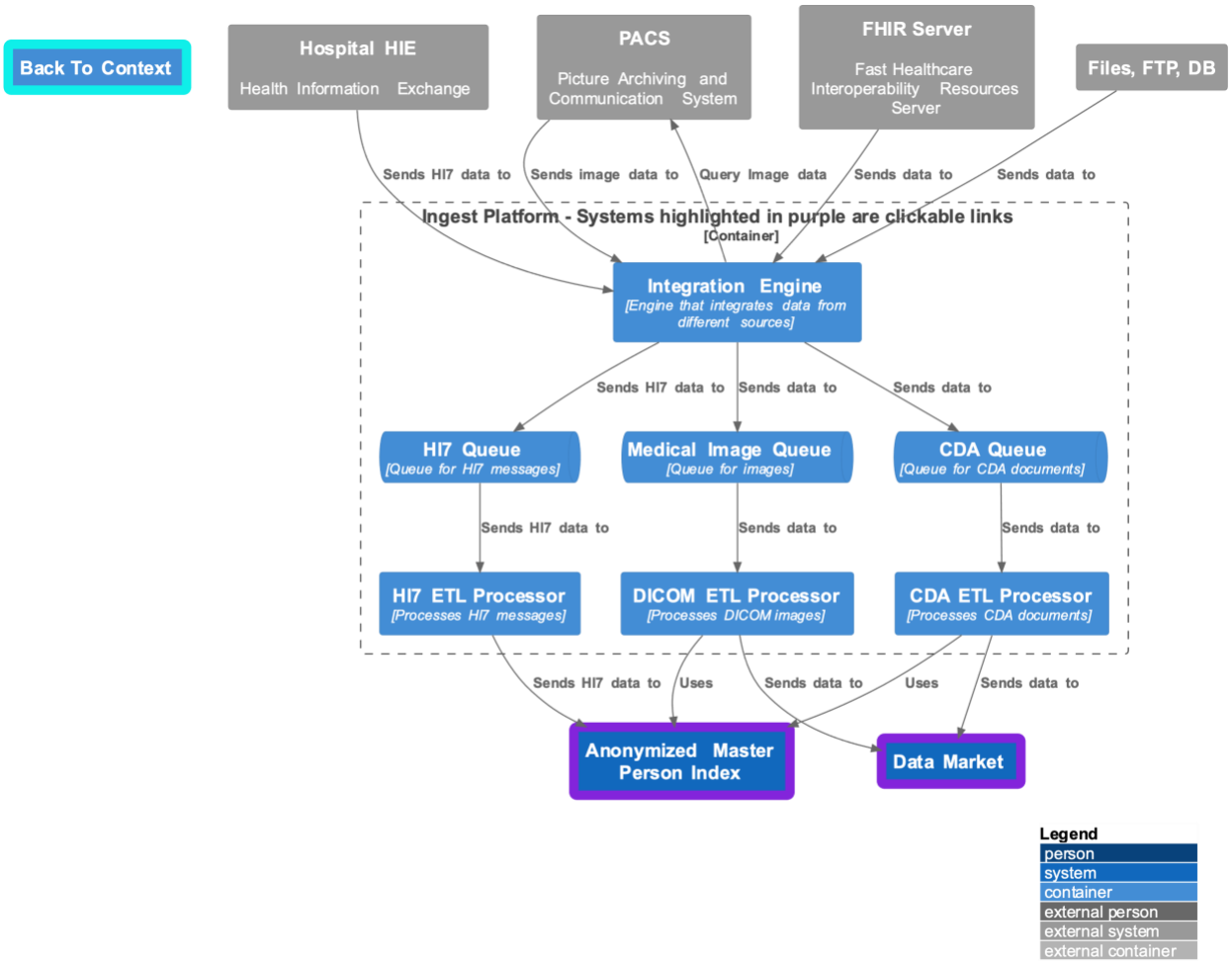
Context Diagram



The tesselate medical research integration platform is composed of 4 main pieces

- Tesselate.io web platform
  - Used to query for data
  - Used by clinicians to annotate data
  - Used by Medical Billers to review earnings from the data they uploaded.
  - Used by Researchers to discover medical data
  - Used by Charge Review specialists to review invoices and make payments.
  - Support analysts to troubleshoot and review logs.
- Ingest Platform
  - Used to ingest data from various sources and de-identify the data.
- Anonymous Master Person Index
  - Creates an anonymized identity that can't be traced back to actual patients
  - Maintains a reference to clinical documents and medical data for anonymized identities to enable viewing the progress of an anonymized patient's condition
- Data Market
  - A set of systems that facilitate the buying, selling, and exchange of data
  - Used by Data Market to aggregate data associated with an identity.

Ingest Platform Container Diagram



The ingest data platform ingests data from various sources, deidentifies it, and passes it to the data market. When data is received in the ETL Processor, it sends the records to the Anonymous Master Person Index to:

1) Find an identity to which to associate the document.
2) If an identity doesn't exist, attempt to create one using the data in the document.

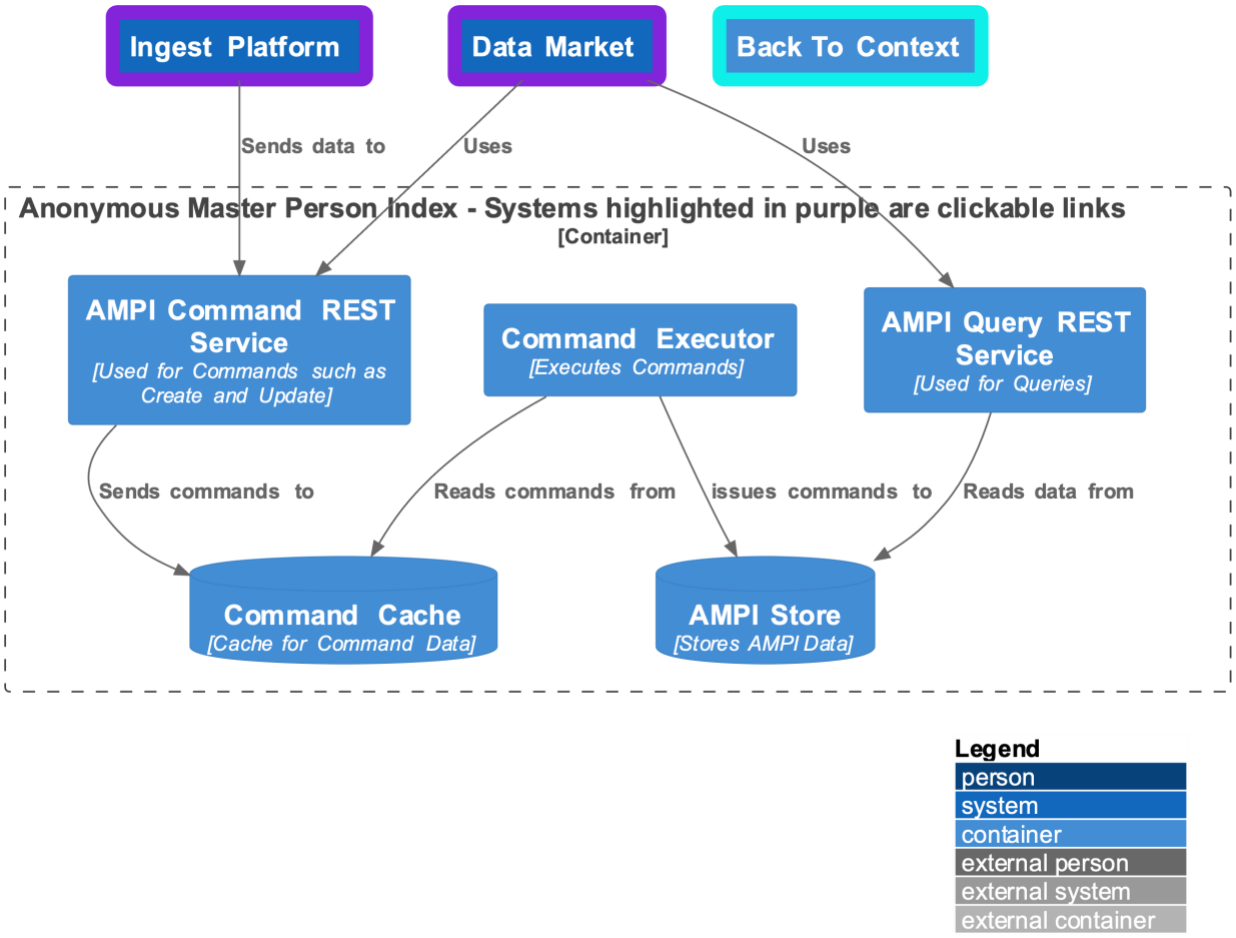
Key Quality attributes

1. **Debuggability and Testability**—The ETL Pipelines are intended to be simple, FIFO-style processors. If an issue arises, developers should be able to look at the queue, replay a message, and see the output captured in the Data Market. They should be able to compare expected vs. actual results to identify and zero in on issues.
2. **Extensibility and Portability**—The processors should use a pipe-and-filter approach. Essential functions should be reusable and extensible by adding

additional sinks. Different hospitals may require specific processing to anonymize a document.

3. **Scalability**—The queue and Processors should scale horizontally. The integration engine license requires us to pay based on the number of deployed instances, so scaling horizontally may cost less than scaling vertically.
4. **Security**—The system is secure at multiple locations. The integration engine has built-in security, LDAP integration, and other features. Data queues will use token-based authentication for reads/writes. The ETL pipelines will read from the service queues but cannot be contacted via an API. The vector of attack with the most significant impact in this system is the queue.

Anonymous Master Person Index Diagram



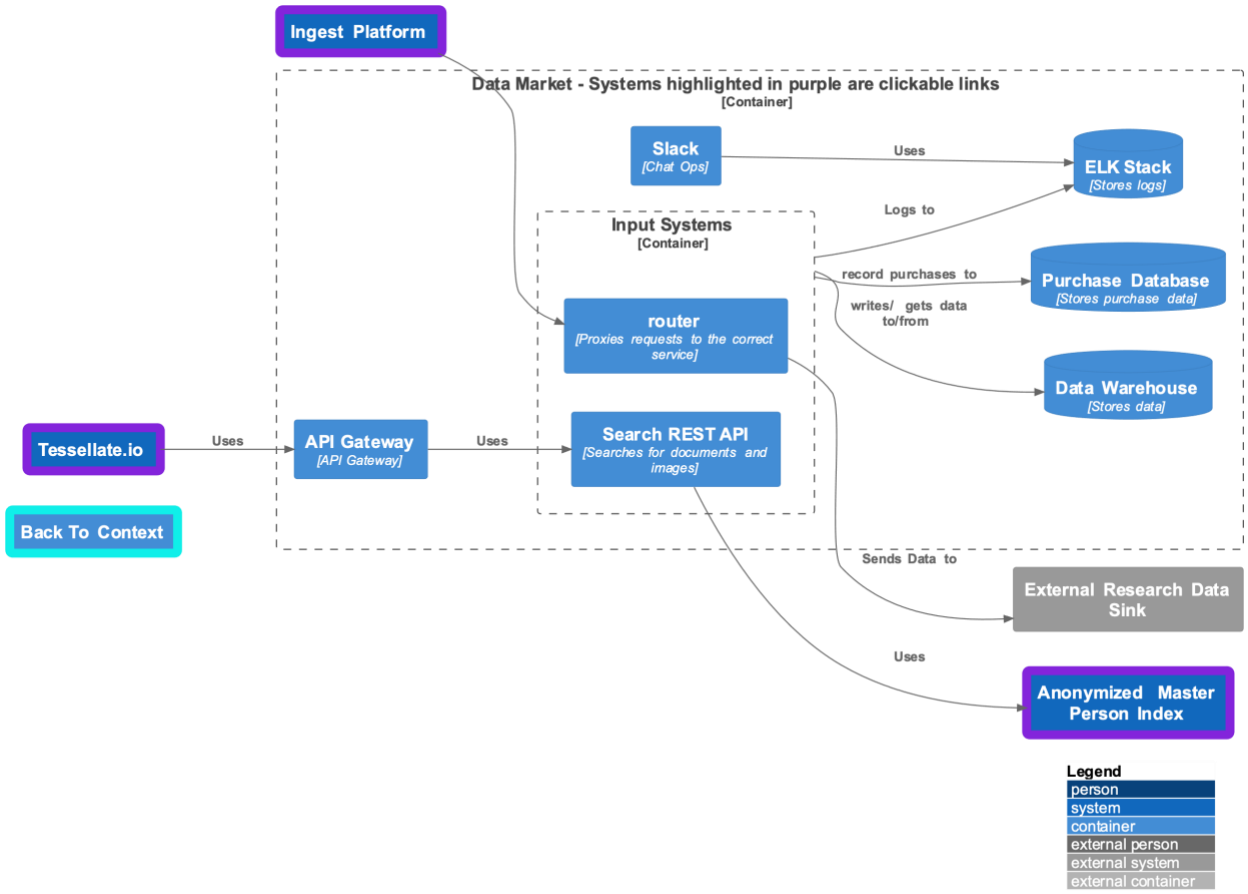
The Anonymized Master Person Index (AMPI) meticulously creates a proxy identity for a patient that can't be traced back to actual patients. This robust feature lets researchers view a patient's condition over time, ensuring utmost da. It creates a cross-reference between a hospital identifier, such as a medical record number, and a unique ID generated for display purposes on the Tessellate side. The hospital identifier is only viewable by clinicians. AMPI supports merging identities for patients in multiple hospital systems with disparate identifiers. It leverages a command query architecture style for better horizontal scalability and domain separation (when required).

Key Quality attributes

- **Debuggability and Testability**—This system allows easy test automation using something like Newman to test the input/ output end-to-end. We essentially issue a command to the Command API and test the result in the Query API.

- **Extensibility**— CQRS offers excellent flexibility, which can contribute to extensibility. We have a clear separation of concerns for reads and writes, so we are free to optimize and evolve models for readability writability and have flexible choices for data storage options depending on how our models/ needs evolve.
- **Scalability**—All components in this system are horizontally scalable. If we have high volumes of writes, we can increase the Command API and Cache instances. If we need higher throughput, we can increase command executor instances. If we are overwhelmed by query instances, we can improve the query REST service instances and add replications for the AMPI store.
- **Security**—The system is secure at multiple locations. Both REST endpoints can be secured via Token-based authorization. They are not directly exposed to the outside and are intended to be used in an internal service-to-service fashion. The command executor cannot be communicated directly; it scans the command cache regularly to perform data operations.

Data Market Container Diagram



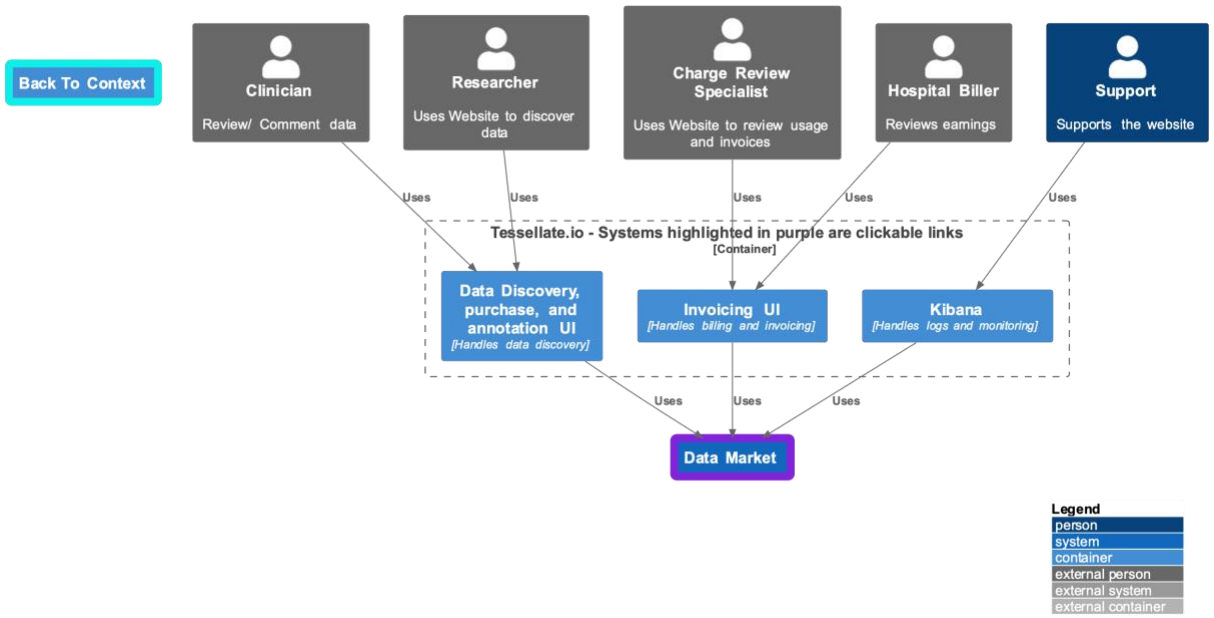
The Data Market facilitates the buying, selling, and exchange of data. The ingest platform brings in anonymized data. The router will send data to the warehouse and stream data to external sinks (i.e., those who subscribe to a data set). All data sent through a sink is logged as a purchase. The tesselate.io domain uses the search REST API to find data. At first, metadata and a preview of a dataset are returned. The user can also purchase the entire dataset. All input systems log to Elk, which Slack uses for alerts.

Key Quality attributes

- **Debuggability and Testability**—The entry point for all operations is throughout input systems, the router, and Search REST APIs. Elk lets us capture the payload and logs as we receive them and whether subsequent operations succeeded or failed. If we couple that with alerting built into Slack for Chat Ops and dashboards via Kibana, it's easy to detect issues and narrow them down to specific components.

- **Extensibility and Portability**—The pipeline is designed so it is possible to add additional external sinks or add stores within our data market. The router uses a rules-based approach that lets us add multiple locations.
- **Scalability**—The router and Search REST API are intended to be horizontally scalable. In contrast, the various data stores in the market are designed to be active-active setups with replications in multiple data centers.
- **Security**—The system is secure at multiple locations. The API gateway validates access from Tesselate.io. The Tokens issued to clients using Tesselate.IO are encrypted, and the API Gateway can decrypt them and validate their signatures and claims. Internal service-to-service communication is secured by a separate token issuer, which uses unencrypted tokens.

Tessellate.IO Container Diagram



The Tesselate.io domain is composed of several UI components. The Data Discovery, Purchase, and Annotation UI allows clinicians to search for and annotate data, and researchers can discover and purchase data sets. The invoicing UI will enable users to review usage-based charges or earnings if they are data providers. Kibana is an open-source data visualization and exploration platform primarily used to analyze log data.