ANSWERS

Note: *the story about the event organisation company was generated by the Author of the work and is fictional*

Event Organisation Company “Event4u”

The company “Event4u” is a leading event provider responsible for delivering large concerts attended by thousands of people. For each performance, the company coordinates artist logistics, VIP guest services, and overall audience and concert management. “Event4u” activities are highly dependent on technologies that ensure real-time efficiency and smooth task performance. The company currently manages various data sources, which include not only traffic camera streams but also overall camera data, and customer data - QR codes scans, security and social media data.

1. The challenge is to identify the most effective way to process this diverse type of data. With traditional batch processing, the company’s data can be stored for later analysis, but this can create a risk of response delays. That happens just because traditional batch processing accumulates a massive amount of data before running computations, which means that the results are only available after the batch is complete. These delays affect the company as it cannot react to live issues and therefore, there is a risk of missing important opportunities or threats. Differently, streaming processing can provide instant analytics, if necessary, but it requires a secure infrastructure and base. There is always a risk of cyberattacks or system failures, which is why a reliable infrastructure is essential.

Let's have a look at the different types of data provided in the case:

Data types and storage table

Data Source	Description	Data type	Storage option	App / tool
Traffic camera feeds	Live video / image streams	Unstructured (photo / video)	HDFS & Kudu / HBase (metadata)	Kafka (data intake), Spark Streaming (process), Apache FLink (analysis), Pig (batch post-event data analysis), PyTorch (anomaly)
QR code scans	Ticket validation at the entrance	Structured (rows: id, name etc.)	Kudu (fast analytics) & Hive (batch history)	Kafka (stream) & Spark & Kudu / Impala
VIP area CCTV	Footage for access / security	Unstructured (photo / video)	HDFS & HBase (metadata)	Kafka & NiFi (streaming)
Audience mobile phones	Presence & incident detection	Semi-structured	HBase, Kudu	Spark streaming & HBase / Kudu
Social media monitoring	Activities: posts, mentions, hashtags	Semi-structured (JSON)	HDFS (archive) & Kudu (real-time sentiment)	Kafka, Spark NLP

Description of the tool / app:

- **Apache Kafka** is a fault-tolerant, distributed streaming platform that can securely receive high-speed data, such as CCTV camera footage, QR scans, and mobile scans.
- **Apache Spark Structured Streaming** supports scalable processing of continuous streams of events in real time; therefore, it can perform transformations, aggregations, and anomaly detection live.
- **Apache Hadoop HBase** provides low-latency random read / write access, storage, making it ideal for maintaining the real-time state of artist, ticket, or audience presence.
- **Apache Kudu** combines fast incremental updates with efficient columnar storage, which allows for creating live dashboards alongside Impala queries.
- **Apache Hive** supports SQL-like querying on historical data stored in HDFS, allowing batch reporting and long-term analysis of situations.
- **Apache Impala** provides massively parallel, low-latency SQL queries directly on Hadoop storage, making it suitable for fast reporting over streaming data in Kudu.

- **Apache NiFi** offers an easy-to-use visual interface for building and routing data flows, it can help when connecting heterogeneous (or diverse origins) data sources like CCTV or other security systems.
- **PyTorch** is a deep learning framework which can analyse video or image streams from CCTV to detect any necessary objects or anomalies live.
- **Apache Spark NLP** brings natural language processing capabilities to Spark pipelines, allowing live sentiment and text analysis on social media streams.
- **Apache Pig** is distributed stream processing engine designed for low-latency, stateful computations with event-time semantics, making it ideal for continuous real-time analytics such as monitoring QR scans or mobile signals.
- **Apache Flink** designed for low-latency, stateful computations with event-time semantics, making it ideal for continuous real-time analytics such as monitoring QR scans or mobile signals.

2. The data generated from attendees' mobile messages (which can be chats, social media etc), in this particular case, are not structured. If we talk in the context of "message" - it can be anything at any size. For example, long text, lots of emojis, photos or videos from the concert. It also depends on the phone model. If it's a modern model, it can include extra attributes like location, or other metadata - from this perspective, metadata can add some structure around the text. However, the overall context has no fixed schema like a database table, especially if different users can send totally different kinds of text. That means the actual message body is unstructured[1] free text. We can conclude that attendees' messages are best described as semi-structured data. The suggestion for the "Event4u" company to use tools like Apache Spark NLP for live extraction of text meaning.

3. Apache Spark Structured Streaming[2] is a fault-tolerant, scalable streaming processing engine, built on Spark SQL, that processes incoming real-time data as a continuously updating table, allowing developers to use familiar dataframe and SQL API's to perform almost instant analytics with strong integration to sources like Kafka, HDFS (explained in the list above), and cloud storage. Spark Streaming can compute and maintain a variety type of data, so the company can gain valuable insight on audience like live crowd metrics, safety / incidents

[1] https://softteco.com/blog/big-data-analytics-in-telecom-industry
[2] https://learn.microsoft.com/en-us/azure/databricks/structured-streaming/concepts

information, social sentiment, etc. From the business metric perspective, it can be described like this:

Type of data	Metrics
People	Attendance, total devices used, entry / exit data, flow direction & activity, social media, complains / happiness, topic trends.
Events	Potential evacuation, crowd pressure or blockage, queue build up, occupancy, safe limits, safety, unathorised entries, CCTV metadata.
Forecast	Ticket sould out, arrivals, top zones in the concert, re-staffing, weather.

By using Apache Spark Structured Streaming, the “Event4u” company can transform raw business data into actionable real-time insights by monitoring crowds, detecting anomalies, or even predicting potential threats. This ensures that events and business in general will run safely, smoothly, and will give the audience a positive experience.

4. An automatic detection of overcrowding in popular zones like restrooms or dining areas is likely possible, as the company already works with live data across events (QR codes, mobile phones, etc.). Those metrics can show the occupancy of the areas or the direction of the crowd flow. Apache Kafka[3] and Spark Structured Streaming can process them in real-time. The work is simple - these tools can count people per special zone or area, measure how fast the crowd is growing, and detect where and when safe limits are reached. Alerts will be sent automatically to a manager or staff with the location of the problem zone (depending on the infrastructure of the event).

Conclusions: Modern tools and technologies allow “Event4u” company to implement an automatic detection of overcrowding by combining data from different sensors and analysing them live, therefore sending alerts in case of safety.

5. A most suitable data managing platform for the “Event4u” would be one that includes real-time opportunities and batch tools. In the table and list above, we have already explored the most suitable tools. In our case, the Author suggests exploring the Apache ecosystem as it provides every efficent tool for the matter of the problem. The company can use Apache Kafka for collecting data, Apache

[3] https://aws.amazon.com/what-is/apache-kafka/

Spark for real-time processing, and HBase or Kudu for fast access. To store historical data, in terms of the Apache ecosystem, HDFS could be a great suggestion. All of the data sources need high security, especially personal data from the tickets; therefore, tools used by the company should have increased security and data protection. Because of the overall risk differences, level of sensitivity, and retention policy, all three sources cannot be stored and secured in the same way.

Conclusions: Each source requires a different storage and security approach. Traffic cameras can be kept briefly with metadata access, QR codes need more encrypted and audited storage as it works with sensitive personal data, and VIP CCTV demands the strict isolation and short retention.

Bibliography:

- https://davidregalado255.medium.com/the-ultimate-guide-to-the-open-source-apache-data-stack-29b4d88f3451
- https://hevodata.com/learn/top-21-hadoop-big-data-tools/
- https://medium.com/expedia-group-tech/apache-spark-structured-streaming-first-streaming-example-1-of-6-e8f3219748ef
- https://researchcomputing.princeton.edu/faq/what-is-a-cluster
- https://softteco.com/blog/big-data-analytics-in-telecom-industry
- https://www.designveloper.com/blog/tree-data-structure/
- https://www.ibm.com/think/topics/structured-vs-unstructured-data
- https://www.ovaledge.com/blog/processing-unstructured-data