Diagramming Challenge for CycleSwift

Supporting materials for the O'Reilly training Threat Modeling Fundamentals - Debug Your Security Design through Whiteboard Hacking by Sebastien Deleersnyder, Toreon.

To learn more on threat modeling, visit https://www.toreon.com/threat-modeling-training/

Exercise Input:

The fictional E-Bike Rental App, "CycleSwift," is an innovative platform that caters to urban residents and tourists seeking convenient and eco-friendly transportation. The app's major stakeholders include the users (riders), the city authorities, and the company operating the app. Users benefit from the app by having on-demand access to electric bikes, allowing them to navigate the city efficiently while reducing their carbon footprint. City authorities are involved as stakeholders due to the app's impact on urban transportation infrastructure, requiring coordination for parking zones and ensuring compliance with local regulations. The operating company, on the other hand, is responsible for maintaining the fleet of e-bikes, managing the app's technology, and customer service. Key use cases involve users locating and unlocking bikes through the app, making payments for rental periods, and accessing customer support for any issues. Additionally, the app provides valuable data analytics for city planning and environmental impact assessments.

User Stories & Use Cases

1. **User Registration & Authentication:**
   - **User Story:** As a new user, I want to sign up and log into the app so that I can start renting e-bikes.
   - **Use Case:** Secure user registration and authentication process.
   - **AWS Components:** Amazon Cognito for user management and authentication.
2. **E-Bike Location & Availability:**
   - **User Story:** As a user, I want to view nearby available e-bikes so I can choose one to rent.
   - **Use Case:** Real-time tracking of e-bike locations and availability.
   - **AWS Components:** AWS IoT Core for device connectivity, Amazon Location Service for geolocation data, and Amazon DynamoDB for storing e-bike status and location data.
3. **E-Bike Unlocking:**
   - **User Story:** As a user, I want to unlock an e-bike through the app so that I can use it.
   - **Use Case:** Secure and seamless e-bike unlocking mechanism.
   - **AWS Components:** AWS IoT Core for secure device communication, AWS Lambda for executing the unlocking logic.
4. **Payment Processing:**
   - **User Story:** As a user, I want to pay for my rental period easily through the app.

- **Use Case:** Secure and efficient payment processing.
- **AWS Components:** AWS Lambda for handling business logic, Amazon API Gateway for RESTful API, and Amazon RDS/Aurora for transactional data storage.

5. **Customer Support:**
   - **User Story:** As a user, I want to access customer support in case of issues or queries.
   - **Use Case:** Efficient handling of customer support tickets.
   - **AWS Components:** Amazon Connect for customer support services, Amazon DynamoDB for ticket management.
6. **Data Analytics for City Planning:**
   - **User Story:** As a city planner, I want to access usage data to improve urban transportation infrastructure.
   - **Use Case:** Data analytics for city planning and environmental assessments.
   - **AWS Components:** Amazon S3 for data storage, Amazon Redshift for data warehousing, Amazon QuickSight for data visualization.

Components for AWS Environment

- **Amazon Cognito:** Manages user registration, authentication, and user pool management.
- **AWS IoT Core:** Connects and manages e-bike IoT devices securely.
- **Amazon Location Service:** Provides maps and location services for tracking e-bikes.
- **Amazon DynamoDB:** NoSQL database for storing real-time e-bike locations, availability, and customer support tickets.
- **AWS Lambda:** Serverless compute service to run the business logic for unlocking bikes, payment processing, and other backend services.
- **Amazon API Gateway:** Creates RESTful APIs to connect the frontend app with backend services.
- **Amazon RDS/Aurora:** Provides relational database services for storing transactional and user data.
- **Amazon Connect:** Cloud-based contact center service for handling customer support.
- **Amazon S3:** Object storage service for storing large amounts of data.
- **Amazon Redshift:** Data warehousing service for analytics workloads.
- **Amazon QuickSight:** Business intelligence service for data visualization and analytics.

Design details

The choice of AWS components is driven by the need for scalability, reliability, and security. For instance, Amazon Cognito provides a robust solution for user authentication, which is crucial for user security. AWS IoT Core is essential for managing the IoT capabilities of the e-bikes, allowing for secure and efficient communication between the app and the e-bikes. DynamoDB offers fast and flexible NoSQL data storage, ideal for the real-time data needs of this application. Lambda and API Gateway form the backbone of the serverless architecture, enabling scalability and reducing operational overhead. Finally, the integration of Amazon Connect, Redshift, and QuickSight ensures that customer support is responsive and data analytics are insightful, supporting both the operational and strategic aspects of the CycleSwift app.

Challenge:
With the provided user stories, use cases, and corresponding AWS components, training participants can begin to sketch out a data flow diagram that visualizes the interactions between these components, and then add 3 trust boundaries you think are important.