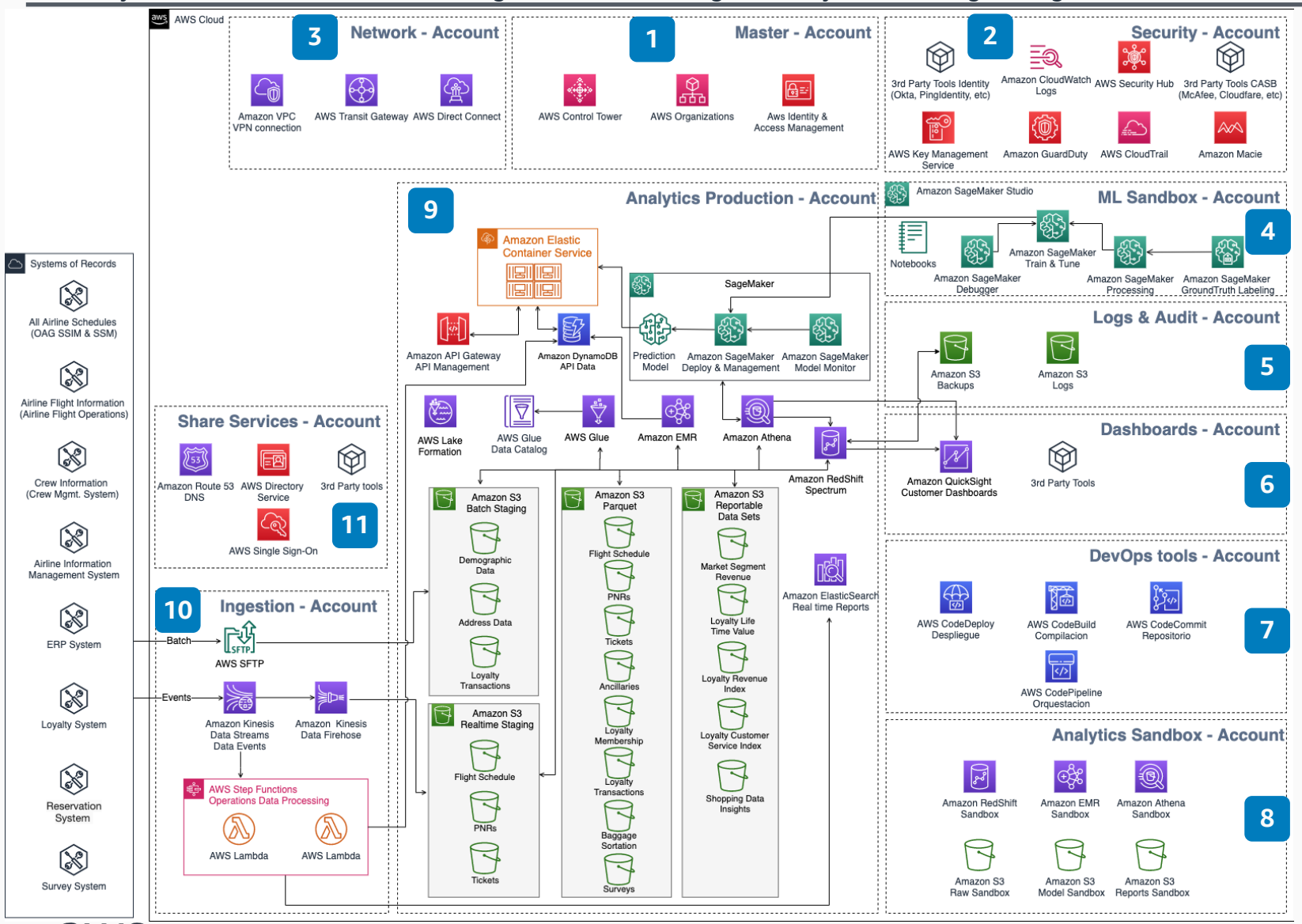
Account and Security Strategy for the Serverless Data Platform

A multi-account strategy is a best practice used to provide resources and security isolation for workloads, and to help with categorization and for reducing blast radius. An analytics architecture demands data governance with high security standards, granting the correct access level to users.



1. Root account in Amazon Web Services (AWS), most critical with restrictive access.
2. Security account for security activities, responsible for maintenance of the overall security posture of all accounts as it scans for vulnerabilities.
3. Centralized network connections from on-premises to share with other accounts. Control communication between accounts with **AWS Transit Gateway**.
4. Enable users to interact with data to develop machine learning (ML) models with the correct security and data access. Publish final models with **Amazon SageMaker**.
5. Centralized logs for monitoring across all accounts to audit activities.
6. Any tools that the customer needs to present data, including third party or AWS tools.
7. DevOps flows to deploy code as a service, and artifacts on the ingestion and analytics accounts.
8. Users who need Dev/Test analytics models with the correct security and data governance.
9. Centralized data services with governance and API management. Run queries against petabytes of data in **Amazon S3** through **Amazon Redshift** Spectrum. Transform data with **Amazon EMR.** Secure repositories with **AWS Lake Formation**.
10. Control ingestion batch or stream that feeds the data lake.
11. Common services to other accounts like golden Amazon machine images (AMIs) or Domain Name Service (DNS).

AWS Reference Architecture