𝗦𝘁𝗮𝗿𝗹𝗶𝗻𝗴 𝗕𝗮𝗻𝗸 𝗔𝗿𝗰𝗵𝗶𝘁𝗲𝗰𝘁𝘂𝗿𝗲 - 𝗡𝗲𝗼𝗯𝗮𝗻𝗸

🔧 Backend Technology Stack:  
  
🔹 Java services in Docker and CoreOS  
🔹 AWS services: EC2, VPC, CloudFormation  
🔹 REST (JAX-RS) for API implementation  
🔹 PostgreSQL as the database  
🔹 Jetty, Guice, Guava, Hystrix frameworks  
🔹 Homegrown libraries for SQL database access, configuration, command line, and background processing  
🔹 Monitoring with Prometheus, log aggregation with ELK, and incident alerting via PagerDuty  
  
🚀 Builds and Deployments:  
  
🔹 Continuous deployment to non-production environments, with a sign-off process for production  
🔹 Automated build, dockerisation, testing, scanning, and deployment completed within an hour  
🔹 Up to 5 code releases to production per day  
🔹 Single stateless service per instance for scalability and resilience  
🔹 Rolling deployments achieved through termination strategy  
  
🏗️ Architecture Principles:  
  
🔹 Microservice architecture for modularity and scalability  
🔹 Perform actions at least once, but at most once  
🔹Asynchronous processing with a focus on idempotence and retry mechanisms  
🔹 Each service constantly working towards data correctness  
🔹 Achieve idempotence through immutability whenever possible  
🔹 No distributed transactions and a mindset of not trusting other services  
  
⚖️ Data Consistency Approach:  
  
🔹 Achieve eventual consistency through microservices - constantly working towards data consistency  
🔹 Use two foundational patterns: Recoverable Command and   
Catch-up Processor  
🔹 Recoverable Command interface processes and marks items as processed using UUIDs  
🔹 Catch-up Processor class selects items and uses Recoverable Command to fix data inconsistencies  
🔹 Context data is stored in the database, while object ID parameters are used for processing  
🔹 Catch-up processing triggered by a scheduler every 5 minutes  
🔹 Ensures microservices are stateless and resilient, allowing safe restarts of EC2 instances  
  
🌟 Additional Information:  
  
🔹 Starling Bank is a cloud-based, mobile-only digital bank in the UK  
🔹 Provides innovative APIs for account opening, payments, loans, etc.  
🔹 Offers a developer platform, custom backend ledger, and integration with Apple Pay and Google Pay  
🔹 No usage of Spring, JEE app servers, Kubernetes, or Terraform  
  
Starling Bank emphasises security, reliability, and resilience in its architecture!   
  
Over to you, what are your thoughts?

