SPRING BOOT · KEYCLOAK · NGINX · TYPESCRIPT

Bank Management System

A secure, role-based core banking platform with Maker–Checker controls

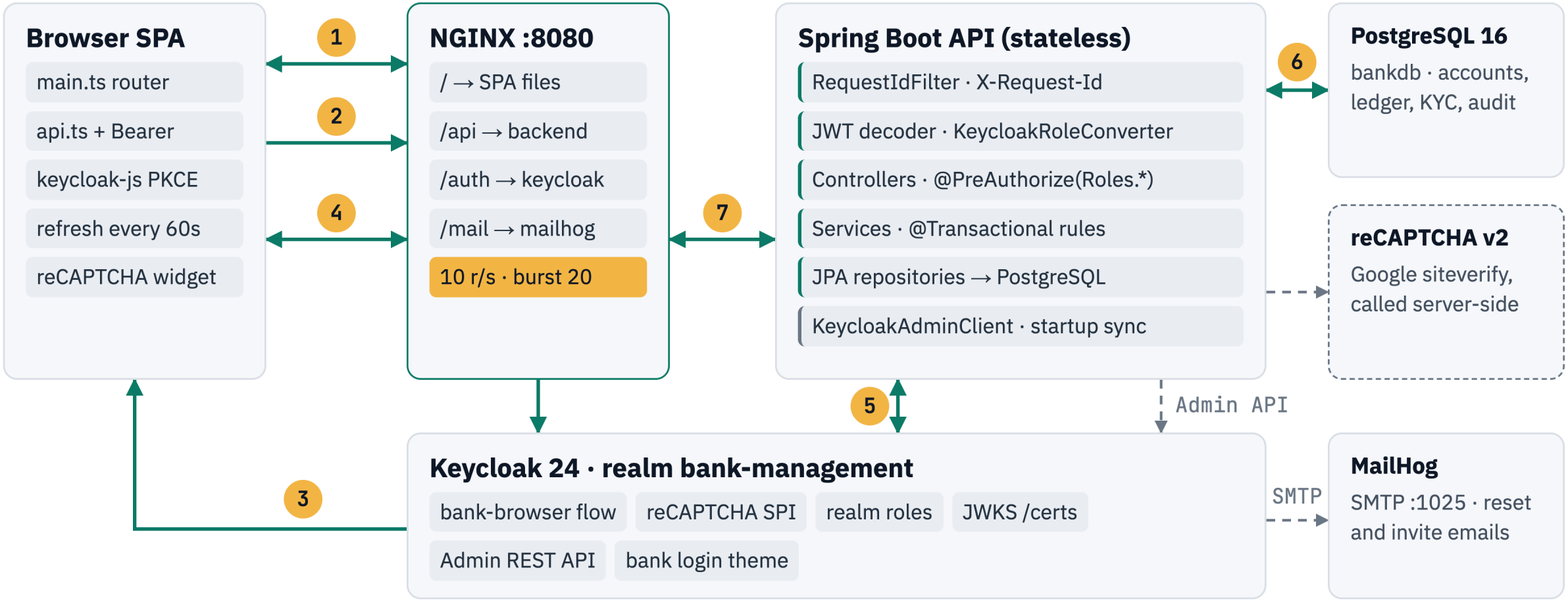
PRESENTED BY

Chandru R · Developer

Technical walkthrough for developers

01 · ARCHITECTURE

How one request travels through the system



1 Load SPA from NGINX
2 Login + reCAPTCHA
3 Auth code → JWT (PKCE)
4 /api call with Bearer token
5 JWT verified via JWKS
6 Rules + SQL, one transaction
7 Audit entry + JSON reply
- - - supporting calls

02 · TECH STACK

Each layer has one job

Layer	Technology	Responsibility
Frontend	Vite 8, strict TypeScript, keycloak-js 24	Role-aware SPA; every API value escaped before rendering
Gateway	NGINX	Single entry point, path routing, rate limiting
Backend	Spring Boot 4, Java 21, Spring Security, JPA	REST API, business rules, OpenAPI docs
Identity	Keycloak 24 + custom Java SPI	OIDC login, realm roles, reCAPTCHA on login
Data	PostgreSQL 16 (H2 in tests)	Customers, accounts, ledger, KYC, audit
Runtime	Docker Compose, MailHog	One-command stack, local email testing

03 · AUTHENTICATION & ACCESS

Every request carries a signed JWT

01 Sign in at Keycloak
OIDC Authorization Code + PKCE, with reCAPTCHA v2 on the form

02 Receive a signed access token
Realm roles travel in the realm_access.roles claim

03 Validate on every API call
Signature checked via JWKS; KeycloakRoleConverter maps roles to ROLE_*

04 Authorize twice
@PreAuthorize on each endpoint, then ownership checks in the service layer

Six realm roles

Role	Can do
ADMIN	Everything; manages staff
EMPLOYEE	Customers and KYC review
MAKER	Raises transaction requests
CHECKER	Approves others' requests
CUSTOMER	Own accounts, payees, consents
TPP	Open Banking APIs, by consent

04 · WORKFLOW

Maker–Checker: two people per money movement

PENDING_APPROVAL

Maker submits

Deposit, withdraw or transfer request. No balance changes.

→

APPROVED

Checker decides

Maker ≠ checker, compared by Keycloak user ID. Or REJECTED.

→

PROCESSING

Engine executes

Row lock taken; limits, frozen and FD rules re-checked.

→

SUCCESS / FAILED

Ledger posted

Transaction and audit entry written. Final states never change.

`@Version`

Optimistic locking stops concurrent status overwrites

`Pessimistic lock`

Approve locks the row, so a request runs only once

`Direct posting`

Only ADMIN, or a CUSTOMER on their own account

05 · SECURITY & RELIABILITY

Defence in depth, layer by layer

reCAPTCHA v2

Login (custom Keycloak authenticator), register and profile update; verified server-side.

Rate limiting

NGINX limit_req: 10 requests per second per IP on /api, burst of 20.

Least privilege

Staff never inherit CUSTOMER; a staff role always overrides it for money movement.

Banking rules

Per-transaction and daily debit limits; frozen and closed accounts blocked; FDs locked.

Audit trail

Money movements and admin actions are recorded with the acting user.

Self-healing startup

Background jobs sync Keycloak (CAPTCHA keys, customer logins, staff roles) with retry.

Thank you

Questions and feedback are welcome.