ACADMI

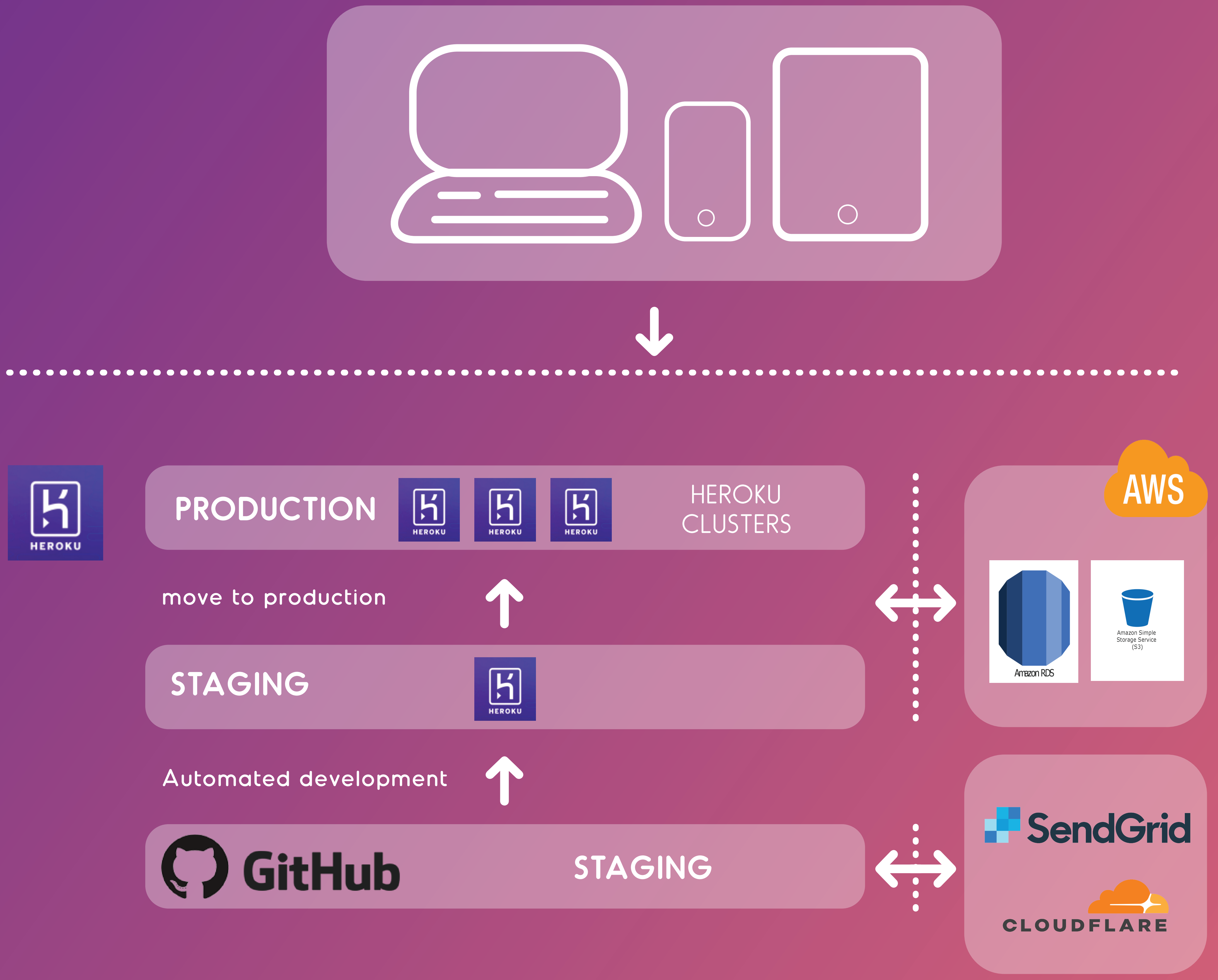
Security Standards & Architectural Map

Acadmi builds cutting-edge technology solutions to enhance training & skills development across multiple industry sectors. With data at the heart of Acadmi's applications, we have the optimal team in place to ensure the security of all of our clients.

ACADMI

Architectural map

RESPONSIVE WEB APP OPTIMIZED FOR DESKTOPS, MOBILE & TABLET



PRODUCTION

HEROKU CLUSTERS

move to production

STAGING

Automated development

STAGING

TECH STACK

RAILS

Rails V5.0

CSS

HTML

- > Automated test suite
- > Continuous integration
- > Test driven development

ACADMI

Security Standards

Error Handling & Logging

- .Display Generic Error Messages ✓
- .No Un-handled Exceptions ✓
- .Suppressed Framework-Generated Errors ✓
- .All Authentication Activities Logged ✓
- .All Administrative Activities Logged ✓
- .Sensitive Data Logged ✓
- .Inappropriate Data not Logged ✓
- .All Logs Stored Securely ✓

Data Protection

- .Consistent Use of HTTPS ✓
- .All requests back to HTTPS Re-Directed ✓
- .HTTP Access Disabled for all Protected Resources ✓
- .Strict Transport-Security Header Used ✓
- .Cloudflare Used to Enhance Security ✓
- .User Passwords Stored Using a Strong, Iterative, Salted Hash ✓
- .Secure Exchange Encryption Keys ✓
- .Secure Key Management Process ✓
- .Valid HTTPS Certificates Used (Cloudflare) ✓
- .Use & Storage of Sensitive Data Limited ✓

Configuration & Operations

- .Application Deployment Automated (Heroku) ✓
- .A Rigorous Change Management Process ✓
- .Regular Design Reviews ✓
- .Regular Code Reviews / Audit ✓
- .Post-Release Security Testing ✓
- .Guidelines for Infrastructure Hardening ✓
- .Incident Handling Plan Defined ✓
- .Internal Team Educated on Security ✓

Authentication

- .Credentials not Hardcoded ✓
- .Strong Password Reset System ✓
- .Strong Password Policy ✓
- .Sign-Up Captcha ✓
- .Denial of Service Attacks Prevented ✓
- .Key Information in Error Messages not Disclosed ✓
- .Database Credentials Stored Securely (No Copies Made) ✓
- .Account Lock-Out Against Brute-Force Attacks ✓
- .Applications & Middleware Run with Minimal Privileges ✓

Session Management

- .Session Tokens Regenerated ✓
- .Random 64-Bit Generator ✓
- .Idle Session Timeout (30 minutes) ✓
- .Absolute Session Timeout (12 hours) ✓
- .Session Invalidated after Logout ✓
- .Logout Button Placed on Every Page ✓
- .Secure Cookie Attributes (HttpOnly & Secure Flags) ✓
- .Cookie Domain & Path Correctly Set ✓
- .Cookie Expiration Time Set ✓

Input & Output Handling

- .Contextual Output Encoded (JSON) ✓
- .Whitelists preferred over Blacklists ✓
- .Parameterised SQL Queries Used ✓
- .Tokens to Prevent Forged Requests ✓
- .Encoding for the App Set ✓
- .Uploaded Files Validated ✓
- .Nosniff Header for Uploaded Content ✓
- .Source of Input Validated ✓
- .X-Frame Options Header Used ✓
- .Secure HTTP Response Headers Used ✓

Access Control

- .Access Control Checks Applied Consistently ✓
- .Gateway Access Control Protocol ✓
- .Principle of Least Privilege Applied ✓
- .Direct Object References for Access ✓
- .Control Checks Not Used ✓
- .Flag on all Users ✓
- .Unvalidated Forwards / Redirects Not Used ✓

Testing / Reviews

- .Regular Code Reviews & Audits ✓
- .Continuous Integration ✓
- .Penetration Testing ✓
- .Load Testing ✓