CLOUD, DEVOPS AND AI ENGINEERING

Four projects, built and run in production.

Authecity Systems LLC is a Minnesota company that migrates, automates and runs cloud infrastructure, and builds the software that sits on top of it. This sheet covers two products we built and operate ourselves, and two client projects. Each was taken from design to a running service.

01 **knownAs.dev** — OUR PRODUCT · PUBLIC BETA
Internet identity for AI agents. Serverless on AWS, defined in Terraform.

02 **Authecity** — OUR PRODUCT · LIVE
Product verification for fast-moving consumer goods. Web app, API and database on AWS.

03 **Youth ministry hybrid hosting** — CLIENT · NON-PROFIT
A cloud-hosted subdomain joined to a church's existing website.

04 **UK platform migration** — CLIENT · MODERNISATION
A WordPress recruitment platform moved off shared hosting onto AWS.

Cloud migration
Off shared hosting and ageing servers, onto environments that can be rebuilt from source.

Delivery automation
CI/CD, infrastructure as code and monitoring, so a release is routine.

AI integration
Model-backed features inside existing software, with a cost ceiling and failure paths.

femi.alla@authecitysystems.com · +1 612 800 1991

KNOWNAS.DEV · PUBLIC BETA

knownAs.dev: internet identity for AI agents.

An AI agent can act on the internet, but it has no name of its own and nothing that says who answers for it. knownAs.dev gives it one: a name like `alice.knownas.dev`, with HTTPS, a public record of who is accountable for the agent, and a route to wherever it runs.

ROLE	SURFACE	HOSTING	STATUS
Product, architecture, build, operations	API, web console, MCP server	AWS, serverless	Public beta at knownas.dev

What it does

- Gives each agent a name, plus one per service, routed over HTTPS to the agent's own server once the owner proves control of it.
- Publishes a public record of what the agent is, who answers for it, and whether it is in good standing.
- Publishes the agent's public key for sites that check Web Bot Auth signatures. The agent keeps its private key.
- Records every change in an append-only audit trail. Agents use scoped, expiring API keys and never hold DNS credentials.

How it was built

- DNS, certificate and email providers sit behind interfaces, so no vendor is wired into the business logic.
- Destructive DNS actions are guarded, and abuse controls were designed in from the first release. Dev and prod are separate, and every design decision is written down.
- The record's format is an open specification, published as an IETF Internet-Draft.

```
$ curl https://alice.knownas.dev/
  .well-known/agent-identity.json
{
  "name": "Alice",
  "fqdn": "alice.knownas.dev",
  "status": "active",
  "owner": {
    "standing": { "status": "active" },
    "verifications": [ { "type": "email" } ]
  },
  "version": "1"
}
```

A real public record, shortened. Anyone can look one up.

Architecture

Python 3.13 functions on AWS Lambda behind API Gateway, with DynamoDB for state. CloudFront serves each agent's name, and CloudFront Functions route service names to the agent's server. Route 53 and ACM handle DNS and wildcard certificates. EventBridge and SQS run scheduled and queued work, SES sends mail, S3 with Object Lock keeps audit archives, and CloudWatch alarms report through SNS. All of it is defined in Terraform and deployed by GitHub Actions.

`Lambda` `API Gateway` `DynamoDB` `CloudFront` `Route 53` `ACM` `SES` `EventBridge` `SQS` `S3 Object Lock` `CloudWatch` `Terraform` `FastAPI` `React` `GitHub Actions`

AUTHECITY.COM · LIVE

Authecity: proving a product is real before someone buys it.

Counterfeit fast-moving consumer goods are common in many markets, and a shopper has no way to tell a genuine unit from a fake before paying. Authecity gives every registered unit a code a shopper can check in seconds, and gives the brand a record of where and when its products are being checked.

ROLE	SURFACE	HOSTING	STATUS
Architecture, build, operations	Web app on desktop and mobile	AWS, us-east-1	Live at authecity.com

For shoppers

- Scan a unit's QR code with the phone camera, or type its serial number. No account needed.
- An immediate answer: registered with the brand, or not found.

For brands

- Register products and issue a serial number and QR code for each unit.
- See where and when products are checked.
- Get alerts when one code is checked repeatedly or in places far apart, which are signs of cloned codes.

Operations

- Deploys run from GitHub Actions. The database is backed up automatically before every migration.
- Sign-up is protected by a human check, and scheduled jobs send an hourly alert digest.

Architecture

A Laravel 12 API on PHP 8.3 with token authentication and role-based permissions for brands, shoppers and administrators. A React 19 and TypeScript web client scans QR codes in the browser. Application, storage and database are separate tiers: the app runs on EC2 behind nginx with TLS, MySQL runs on Amazon RDS, and QR images and brand logos live in S3. Consumer-facing checks stay fast while brand reporting runs behind them.

`Laravel 12` `PHP 8.3` `React 19` `TypeScript` `EC2` `RDS MySQL` `S3` `nginx` `Cloudflare Turnstile` `GitHub Actions`

WHY IT MATTERS TO CLIENTS

It is the same tiered architecture we set up for client applications, run by us in production. When we recommend a design, it is one we operate ourselves.

03 · NON-PROFIT · MINNESOTA

Youth ministry hybrid hosting

The problem

A church's main website ran on traditional shared hosting. Its youth ministry needed a site of its own that was faster and more reliable, with HTTPS, on a small budget, and without moving or disturbing the main site.

Result

An independent ministry site on its own infrastructure, secured with HTTPS, while the parent site stayed exactly where it was.

What we did

- Built a dedicated WordPress environment on a hardened AWS instance.
- Joined it to the church's domain as a subdomain, routing only that name to AWS.
- Used a managed database, object storage for media and backups, and automated certificates.
- Added log monitoring and uptime visibility, sized to keep running costs low.

EC2 RDS S3 CloudWatch DNS TLS WordPress

04 · MODERNISATION · UNITED KINGDOM

UK platform migration

The problem

A UK education recruitment platform ran a customised WordPress site on shared hosting. Pages loaded slowly, the database struggled, there was no reliable backup, and the hosting gave no control over server configuration.

Result

Faster, more dependable pages, automated backups where there had been none, and full control of the server for further development.

What we did

- Moved the application to a Linux server on AWS running nginx and PHP-FPM, tuned for its workload.
- Moved the data to a managed database, and media and automated backups to object storage.
- Planned the DNS cutover to keep downtime to a minimum.
- Added logging, metrics and alerting.

EC2 RDS S3 CloudWatch nginx PHP-FPM WordPress