AI Systems Engineering Syllabus

~9 months

Acceleration Program

System design

Table of Contents

Program Overview

This program is designed to help experienced engineers move from feature-level work to system-level engineering. You'll learn to design, build, and operate real-world systems — from distributed architectures to production AI pipelines — and finish with one cumulative, production-grade system that demonstrates your engineering capabilities end to end.

~9 mo	8	1
part-time · 40 weeks	modules + capstone	production system you own

Who This Program Is For

Built for working engineers with roughly 5–10+ years of software experience. An entry assessment keeps the cohort at the right level — there are no complete beginners in the room.

TARGET ROLES

- AI Systems Engineer
- AI Systems Architect
- Forward Deployed Engineer, Applied AI
- AI Engineer, AI Foundation & Tooling
- Forward Deployed Engineer, Ecosystem
- Software Engineer, ML Infrastructure
- Production Engineer
- Software Engineer, ChatGPT Infrastructure
- Software Engineer, AI Reliability

Program at a Glance

The program is tuned to a length one part-time engineer can finish without losing the breadth of the original scope. The Operator UI workshop is folded into the modules where its tools are used.

Module 1 — Foundations of System Engineering — 2 weeks
Architecture principles, observability, distributed-systems mental models

Module 2 — API & Service Architecture — 3 weeks
REST, gRPC, service boundaries, domain-driven thinking

Module 3 — Data Architecture & Storage Patterns — 4 weeks
SQL/NoSQL, replication, sharding, Redis caching

Module 4 — Cloud Infrastructure & Platform Engineering — 8 weeks
Kubernetes, Terraform, CI/CD, observability — the heart of the program

Module 5 — Distributed Systems & Event-Driven Architecture — 4 weeks
CAP, idempotency, resilience patterns, Kafka

Module 6 — Security & Compliance Architecture — 4 weeks
Threat modeling, managed auth, secure pipelines, compliance mapping

Module 7 — AI Systems Design & Integration — 6 weeks
LLM integration, RAG, agentic workflows, evaluation & AI observability

Module 8 — Architecture Documentation & Leadership — 2 weeks
C4 diagrams, ADRs, technical communication — woven throughout

Capstone — one production-grade system — 7 weeks
One cumulative system taken to a production Definition of Done

Total program length — 40 weeks

Portfolio-ready projects. Across the program you build production-grade systems — a monitoring platform, event pipelines, an internal developer platform, and an LLM gateway — real, defensible work you can showcase in your portfolio and walk through in interviews.

How You'll Practice

Production skills need real traffic, real consumers, real failures, and real business context. The program supplies all four through a concrete simulated environment — not scenario prose. Every module's project modifies one real, runnable system.

1

A Running Reference Platform

Engineers are handed a working multi-service application — repo, infra, and seeded data with a short business brief. Every module's project modifies this one real, runnable system.

2

Synthetic Traffic

Load generators (k6 / Locust) drive realistic request patterns, producing real latency, throughput, and error data — and making SLOs measurable without real users.

3

Provided Consumers

"Don't break existing consumers" is enforced by contract-test suites that act as the existing users: an API change that breaks the contract fails the build, like breaking a paying customer.

4

Scripted Failure

Engineers experience failure rather than read about it: kill-a-pod and drain-a-node labs, consumer crashes and replay, seeded vulnerable builds, injected provider outages — all scripted and graded.

5

Anchored Business Context

Each module's scenario brief is short and attached to the platform engineers can touch — context comes through the system, not paragraphs of fiction.

Module 1: Foundations of System Engineering

2 WEEKS

Engineers step back from feature work and learn to see whole systems: how the pieces of a modern product fit together, where systems break under load, and how to reason about scale before writing a line of code. This module sets the mental models the rest of the program builds on.

Topics

- Modern Software Architecture
- Monolith vs Microservices
- System Design Principles
- Distributed Systems Fundamentals
- Back-of-Envelope Estimation
- System Diagrams & C4 Vocabulary

Skills You'll Gain

- Understanding Architecture Tradeoffs
- Identifying System Bottlenecks
- Reasoning About Scalability
- Capacity Estimation

Builds toward · Read the blueprints — a full analysis of the reference platform you build all program.

Module 2: API & Service Architecture

3 WEEKS

Every system the target companies run is a set of services talking to each other. Engineers learn to design clean, evolvable APIs, draw sensible boundaries between services, and choose the right communication style — the difference between a system that grows gracefully and one that collapses under its own integrations.

Topics

- REST Architecture
- API Versioning & Evolution
- gRPC
- GraphQL
- Service Boundaries & DDD
- Sync vs Async Communication
- API Gateways & Rate Limiting

Skills You'll Gain

- Designing Evolvable APIs
- Implementing gRPC Services
- Decomposing Monoliths
- Choosing Communication Styles

Builds toward · The platform's front door — the public REST API and one internal gRPC service.

Module 3: Data Architecture & Storage Patterns

4 WEEKS

Systems live and die by their data layer. Engineers learn to pick the right database for the job, keep it fast as data grows, and use caching without corrupting correctness — the decisions that show up in every infrastructure interview.

Topics

- Relational Fundamentals
- NoSQL Families
- Replication & Read Replicas
- Partitioning & Sharding
- Redis Caching
- Cache Consistency & pgvector

Skills You'll Gain

- Selecting Storage Models
- Implementing Replication
- Indexing From Query Plans
- Caching Without Corruption

Builds toward · The platform remembers — the Postgres + replica + Redis data layer, with ADRs.

Module 4: Cloud Infrastructure & Platform Engineering

8 WEEKS THE HEART OF THE PROGRAM

This is the module the target jobs are made of. Engineers take everything they've built and learn to run it the way real companies do: in containers, on Kubernetes, provisioned by code, shipped by pipelines, and watched by dashboards. “It works on my machine” becomes “it runs in the cloud, deploys itself, and pages me when it's sick.”

Topics

- Docker & Containers
- Kubernetes on EKS
- Terraform & IaC
- CI/CD Pipelines
- Observability: Prometheus / Grafana / OTel
- AWS Cloud Architecture
- GitOps (ArgoCD / Flux)
- FinOps & Platform Engineering

Skills You’ll Gain

- Containerizing Services
- Deploying to Kubernetes
- Provisioning With Terraform
- Building CI/CD Pipelines
- Instrumenting & Alerting

Builds toward · The platform goes live — containerized, provisioned, deployed, and instrumented end to end.

Module 5: Distributed Systems & Event-Driven Architecture

4 WEEKS

Once a system is many services, new failure modes appear: messages arrive twice, halves of the system disagree, one slow dependency stalls everything. Engineers learn the patterns that keep multi-service systems correct and resilient — and event-driven design with Kafka, the backbone of modern data flow.

Topics

- CAP & Consistency Models
- Resilience Patterns
- Idempotency & Outbox
- Sagas & Distributed Transactions
- Event-Driven Architecture
- Kafka
- Consensus & CRDTs

Skills You'll Gain

- Reasoning About Consistency
- Building Idempotent Consumers
- Designing Event Pipelines
- Surviving Partial Failure

Builds toward · The platform reacts — an event-driven workflow that's idempotent and replay-safe.

Module 6: Security & Compliance Architecture

4 WEEKS

A live platform is a target. Engineers learn to think like an attacker (threat modeling), lock down identity and secrets the way real companies do, secure the delivery pipeline itself, and understand the compliance regimes — GDPR, SOC 2, HIPAA — their employers are audited against.

Topics

- Threat Modeling (STRIDE)
- Zero-Trust Principles
- OAuth 2.0 / OIDC
- Secrets Management
- Supply-Chain Security: Semgrep / Trivy
- TLS / HTTPS
- Compliance by Design

Skills You'll Gain

- Threat-Modeling Architectures
- Securing Identity & Secrets
- Hardening Pipelines
- Mapping Compliance

Builds toward · The platform locks its doors — auth, RBAC, secrets, pipeline scanning, and audit logging.

Module 7: AI Systems Design & Integration

6 WEEKS THE DIFFERENTIATOR

This is where the program earns the "AI" in its title. Engineers integrate LLMs into the platform the way the target companies do: a reliable gateway, retrieval over real data, an agentic workflow with guardrails, and — most importantly — evaluation and AI-specific observability, because an AI feature you can't measure is a liability. They also wire AI into the development lifecycle itself.

Topics

- LLM Fundamentals
- Production LLM Integration
- RAG Architecture
- Agentic Frameworks (LangGraph)
- Evaluation Pipelines
- AI Observability (Langfuse)
- ML-Platform Exposure
- AI Governance

Skills You'll Gain

- Building LLM Gateways
- Shipping Production RAG
- Orchestrating Agents Safely
- Evaluating & Tracing AI

Builds toward · The platform gets a brain — gateway, RAG, an agentic workflow, eval-gated CI, and AI dashboards.

Module 8: Architecture Documentation & Leadership

2 WEEKS WOVEN THROUGHOUT

Senior engineers are paid for judgment they can communicate. Engineers document the system they've built the way staff engineers do — C4 diagrams, decision records, a real architecture proposal — and practice defending it to technical and non-technical audiences. This module doubles as capstone preparation.

Topics

- C4 Model & ADRs
- Tradeoff Analysis
- Stakeholder Communication
- AI-Assisted Documentation

Skills You'll Gain

- Documenting Architecture
- Writing Architecture Proposals
- Defending Decisions

Builds toward · The capstone proposal — defended before a mixed panel.

CAPSTONE · 7 WEEKS

One System, Production-Grade

Engineers take their assembled platform — or a variant use case of their choosing — the last mile against a production Definition of Done, then present and defend it before a panel of reviewers. One cohesive artifact behind the certificate, instead of four startup-scale products.

- SLOs held under injected failure
- Eval-gated CI
- Full observability
- Secured: auth, secrets, scanning, audit log
- IaC-provisioned
- Documented with C4 + ADRs
- Defended before a panel

The AI in Practice Thread

Woven through every module with no extra weeks. Each module carries at least one graded AI-native moment: produce a real deliverable with an AI assistant — scaffold a service, generate Terraform, draft ADRs, write tests, triage an incident — then find and fix what the AI got wrong. This mirrors the AI-first development workflow target employers now expect.

What You'll Work With

The full stack you'll design, build, and operate — grouped by the areas of the system you'll own.

Cloud, infrastructure & platform

Docker
Kubernetes (managed)
Terraform / IaC
CI/CD
AWS
GitOps (ArgoCD / Flux)
FinOps
Backstage

APIs, services & data

REST + versioning
gRPC
GraphQL
Domain-driven boundaries
Postgres + Redis + replication
Sharding
Caching
pgvector
Kafka
Event-driven architecture
Feature stores

Distributed systems & reliability

CAP & consistency
Resilience patterns
Sagas
Observability (Prometheus / Grafana / OpenTelemetry)
SLOs
Consensus
Multi-region

Security & compliance

Threat modeling (STRIDE)
Managed auth (OAuth / OIDC)
Secrets
Pipeline scanning
Zero-trust
TLS
Compliance (GDPR / SOC 2 / HIPAA)

AI systems

LLM integration
RAG
Agentic workflows
Evaluation & AI observability
AI governance
A/B & shadow
Model serving (vLLM / Ray)
AI-first dev workflow

Architecture & engineering practice

C4 diagrams & ADRs
Tradeoff analysis
Capacity estimation
Stakeholder communication

Design the system.
Build the system.