Here’s how the **Backend Engineer’s** responsibilities would look in your Threat Intelligence Platform, based on the expanded phase map and your team setup.  
Their main role is to **bridge the data, AI logic, and user-facing layers** — building secure, scalable APIs and backend services that power the Flutter app, dashboards, and integrations.

**Phase 0 – Platform Architecture & Infrastructure Setup**

* Design **backend architecture** (microservices vs monolith, API gateway, load balancing).
* Choose backend tech stack (e.g., FastAPI, Django, Node.js, Go).
* Set up **CI/CD pipelines** for backend deployment.
* Work with Security Engineers to define **authentication, authorization, and API security**.
* Ensure backend infrastructure is **scalable and highly available**.

**Phase 1 – Data Acquisition & Ingestion**

* Develop **ingestion API endpoints** for manual IOC submission (analyst input, partner data sharing).
* Manage **webhook listeners** for real-time feed pushes.
* Expose backend services for **triggering data ingestion jobs** managed by the Data Engineer.
* Handle **API authentication** for third-party threat feed integrations.

**Phase 2 – Data Normalization & Storage**

* Build backend services that interface with storage layers (hot/cold databases, Elasticsearch, object stores).
* Implement **query layers** for normalized data retrieval.
* Provide **REST/GraphQL endpoints** for Data/AI engineers to fetch cleaned data for model training or inference.

**Phase 3 – Threat Data Management**

* Create APIs for **searching, filtering, and retrieving IOCs** with pagination and filtering options.
* Implement **tagging and classification endpoints** (e.g., attach TLP level to IOC).
* Provide enrichment API endpoints (WHOIS, GeoIP, ASN lookups) and integrate with Data Engineer’s enrichment pipelines.
* Maintain **data versioning APIs** for historical IOC tracking.

**Phase 4 – Analysis & Correlation**

* Build backend orchestration services to run **correlation jobs** (e.g., link IOCs to campaigns).
* Integrate AI Engineer’s **threat scoring models** into backend endpoints.
* Manage graph database queries for **relationship mapping** APIs.
* Expose endpoints for **time-based trend analysis**.

**Phase 5 – Visualization & Reporting**

* Serve **data to dashboards** and the Flutter app via backend APIs.
* Implement **report generation services** (PDF/CSV/HTML export).
* Support **real-time streaming APIs** (WebSockets, SSE) for live dashboard updates.
* Provide **custom query builder APIs** for advanced users.

**Phase 6 – Security Operations Integration**

* Build connectors to send data to SIEM/SOAR platforms (Splunk, QRadar, Elastic SIEM).
* Create backend services for **alert management** (create, acknowledge, resolve alerts).
* Implement **push notification services** for the Flutter app via Firebase or APNs.
* Provide **secure API endpoints** for third-party SOC tools to query threat intelligence.

**Phase 7 – Governance, Compliance & Security**

* Enforce **RBAC (Role-Based Access Control)** at the API layer.
* Implement **audit logging APIs** for all data changes and user actions.
* Work with Security Engineers to integrate **MFA, session management, and API rate limiting**.
* Ensure **compliance in API data handling** (masking sensitive info, filtering based on TLP).

**Phase 8 – AI & Automation (if included)**

* Host and expose AI/LLM inference endpoints from the AI Engineer’s models.
* Build **LLM chat endpoints** for natural language threat queries.
* Manage **async processing queues** for long-running AI tasks.
* Provide APIs for analysts to **give feedback on AI results** (human-in-the-loop learning).

**Expected Outcomes from Backend Engineer’s Work**

* **Reliable, secure, and well-documented APIs** for all platform features.
* **Fast and scalable data access layer** that feeds dashboards, mobile apps, and integrations.
* **Smooth integration** between ingestion pipelines, AI models, and visualization layers.
* **SOC-ready backend services** that push threat data where it’s needed in real time.
* **Strong security controls** at every API and backend entry point.

**UPDATED PLAN:  
Backend Engineer – Responsibilities in TIP Pipeline**

**Phase 0 – Project Initiation & Planning**

* **Define backend service architecture (microservices vs monolith, API gateway, message queues).**
* **Estimate backend resource needs (compute, storage, scaling strategies).**
* **Choose backend frameworks and languages (e.g., Python/FastAPI, Node.js/Express, Go).**
* **Collaborate with Security Engineers on deployment model (on-prem, cloud, hybrid).**

**Phase 1 – Data Acquisition & Ingestion**

* **Implement feed connectors for multiple formats (STIX/TAXII, REST APIs, CSV, RSS, email parsing).**
* **Build ingestion microservices with retry logic, rate limiting, and error handling.**
* **Integrate with message queues (Kafka, RabbitMQ) for scalable ingestion.**
* **Create connector SDKs so new feeds can be added quickly.**
* **Expose ingestion status APIs for monitoring.**

**Phase 2 – Data Normalization & Storage**

* **Implement data cleaning and parsing services:**
  + **Format conversion (CSV → JSON → internal schema).**
  + **Mapping to STIX 2.1/MISP schemas.**
* **Collaborate with Data Engineers on ETL pipelines for structured storage.**
* **Build API endpoints for querying normalized IOCs.**
* **Design database interaction layer (ORM or direct queries) for multiple DB types (Elasticsearch, PostgreSQL, Neo4j).**

**Phase 3 – Threat Data Management**

* **Develop backend modules for:**
  + **IOC tagging & categorization**
  + **Confidence scoring based on feed reliability**
  + **TLP enforcement logic**
* **Build advanced search and filtering APIs (full-text, fuzzy search, date range).**
* **Implement asynchronous enrichment services:**
  + **Integrate with VirusTotal, Shodan, WHOIS, CVE APIs.**
  + **Use background workers (Celery, BullMQ).**
* **Work with AI Engineers to provide enrichment outputs in model-friendly format.**

**Phase 4 – Analysis & Correlation**

* **Implement correlation engines:**
  + **Match IOCs across multiple feeds and historical data.**
  + **Link IOCs to campaigns, actors, and TTPs.**
* **Develop threat scoring API for frontends & SOC tools.**
* **Integrate AI-assisted modules from AI Engineers (ML anomaly detection, NLP summaries).**
* **Store correlation results for historical review.**

**Phase 5 – Visualization & Reporting**

* **Build backend endpoints for dashboards, graphs, heatmaps.**
* **Create APIs to serve:**
  + **Relationship graph data (actor ↔ IOC ↔ campaign).**
  + **Geolocation data for attack mapping.**
* **Implement report generation services (PDF, HTML, JSON, CSV).**
* **Provide data aggregation endpoints for Flutter apps and web dashboards.**

**Phase 6 – Security Operations Integration**

* **Develop SIEM/SOAR integration APIs (Splunk, QRadar, Cortex XSOAR).**
* **Implement alert notification services (Email, Slack, Teams, PagerDuty).**
* **Create incident response API hooks:**
  + **Push IOCs to firewalls, EDRs, WAFs.**
* **Work with Security Engineers to enforce secure API access (JWT, OAuth2, RBAC).**

**Phase 7 – Testing & QA**

* **Write unit & integration tests for backend services.**
* **Build mock feed generators for testing ingestion.**
* **Collaborate in load testing for high-volume feed ingestion.**
* **Work with Security Engineers for API penetration testing.**

**Phase 8 – Governance, Compliance & Security**

* **Enforce RBAC in backend services.**
* **Implement audit logging for API calls and data changes.**
* **Support compliance requirements (GDPR, ISO 27001).**
* **Integrate with vault services for secure API key and credential storage.**

**Phase 9 – Deployment & Release Management**

* **Set up CI/CD pipelines for backend services.**
* **Manage staging vs production deployments.**
* **Implement blue/green or canary deployments for API changes.**

**Phase 10 – Monitoring & Continuous Improvement**

* **Implement backend monitoring (Prometheus, Grafana).**
* **Log API performance, ingestion success rates, and error counts.**
* **Continuously optimize API response times and data query performance.**

**Phase 11 – Documentation & Training**

* **Write API documentation (Swagger/OpenAPI).**
* **Provide developer guides for integrating with backend APIs.**
* **Train SOC analysts and Flutter developers on how to use backend endpoints.**

**Collaboration Points**

* **Data Engineers – Work on ingestion/normalization pipelines and DB schema design.**
* **AI Engineers – Expose clean APIs for model input/output, integrate ML/LLM outputs into backend.**
* **Flutter Developers – Deliver API endpoints optimized for mobile data usage.**
* **Security/Network Engineers – Secure APIs, handle authentication, ensure compliance.**