Efficient Encoding Platform for Complex Encoding Workflows

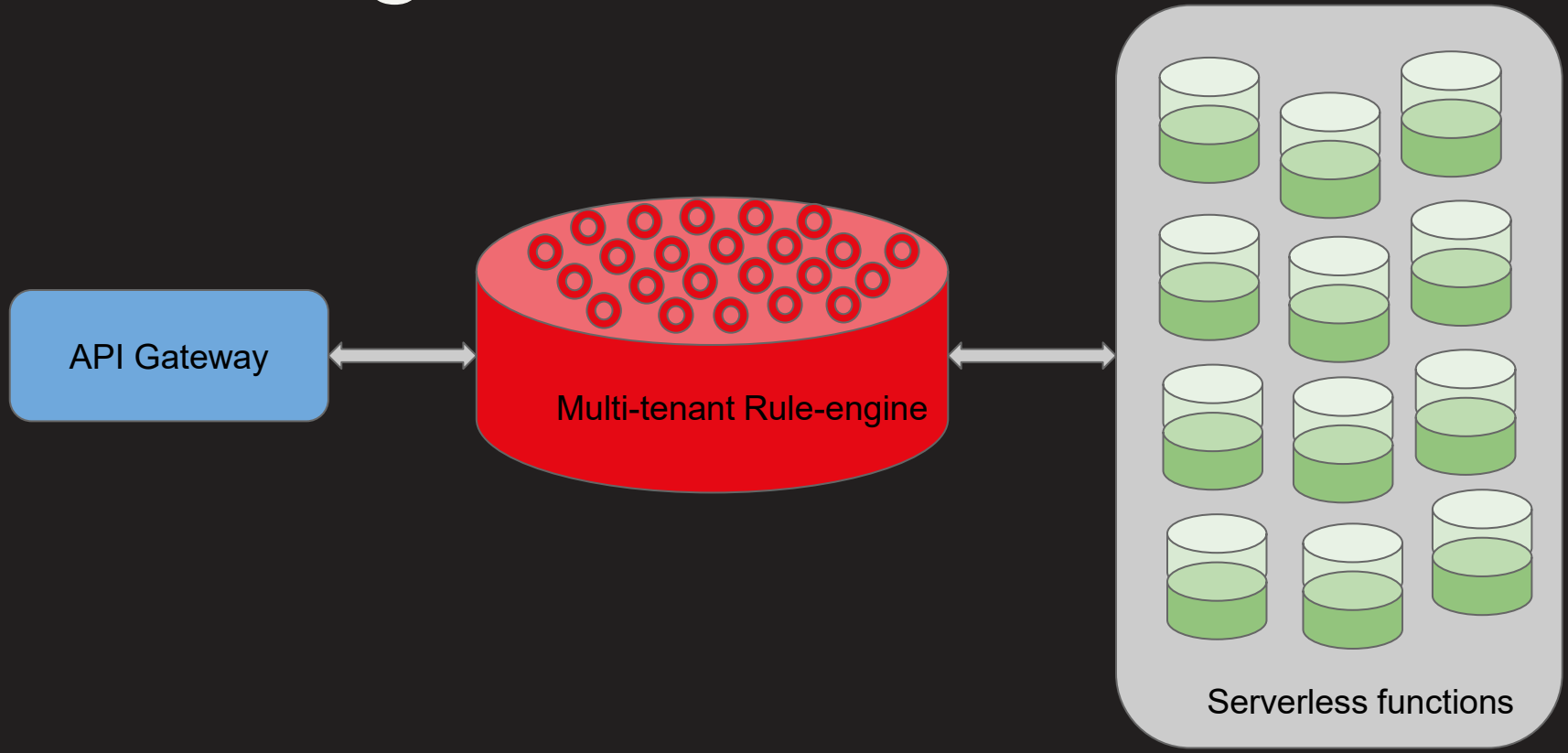
NETFLIX

Sujana Sooreddy
Poorna Reddy

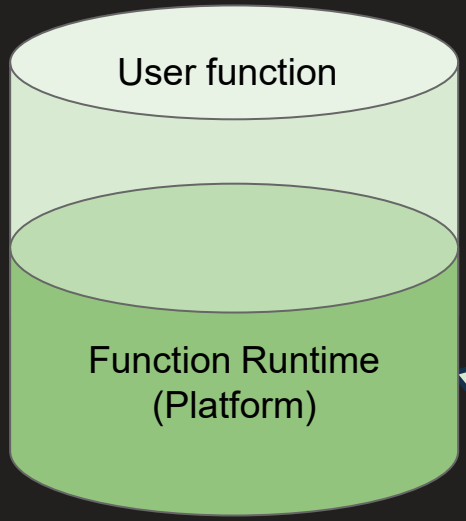
Background

- First generation encoding pipeline went live with streaming launch in 2007
- Second generation pipeline added scale but was difficult to operate
- Third generation, Reloaded, it served us for 8 years, stable and massively scalable.
    - Designed for a small team of developers
    - monolithic architecture
    - Centralized data model
    - 2 weeks release train
- Currently, the fourth generation encoding platform.

Encoding Platform

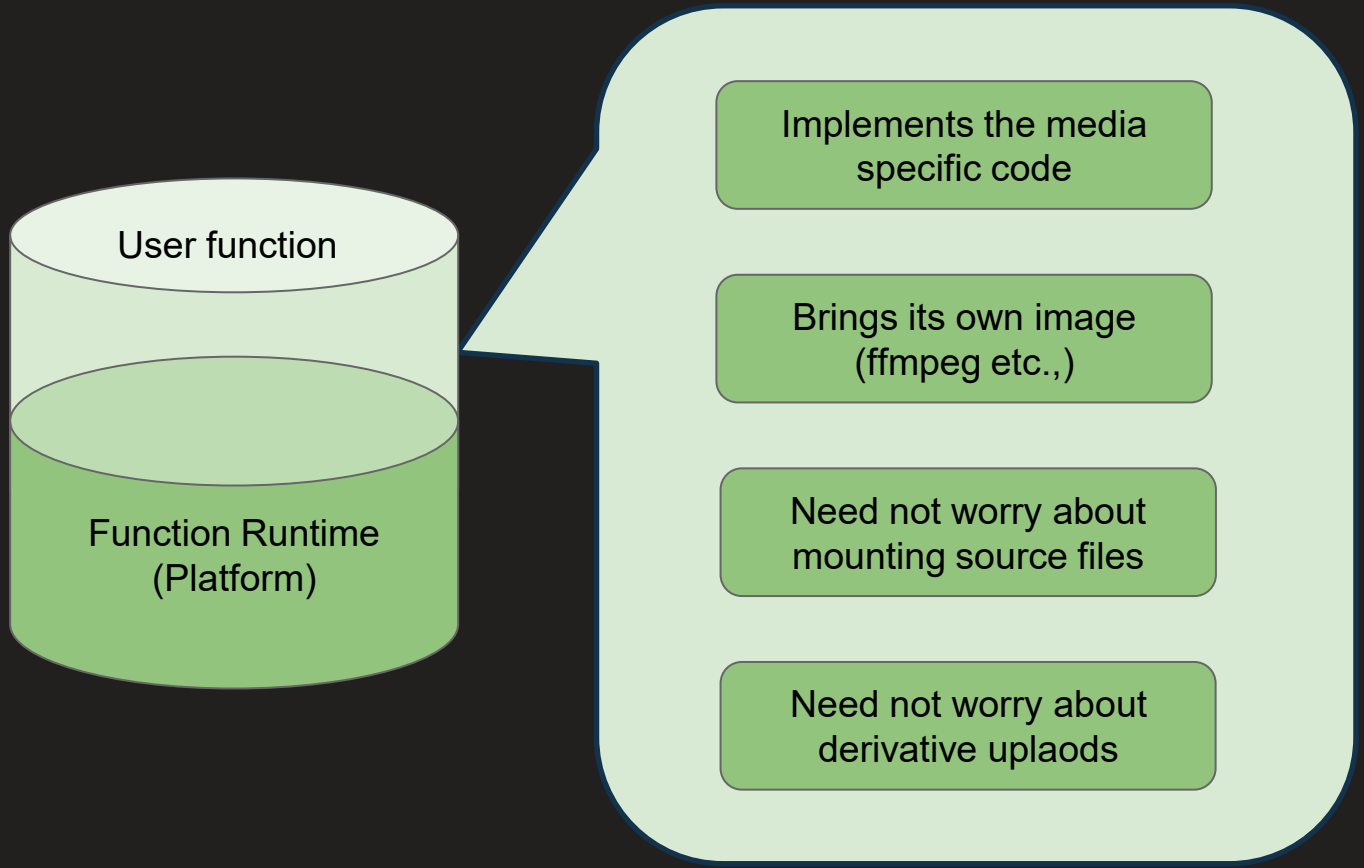


Serverless functions

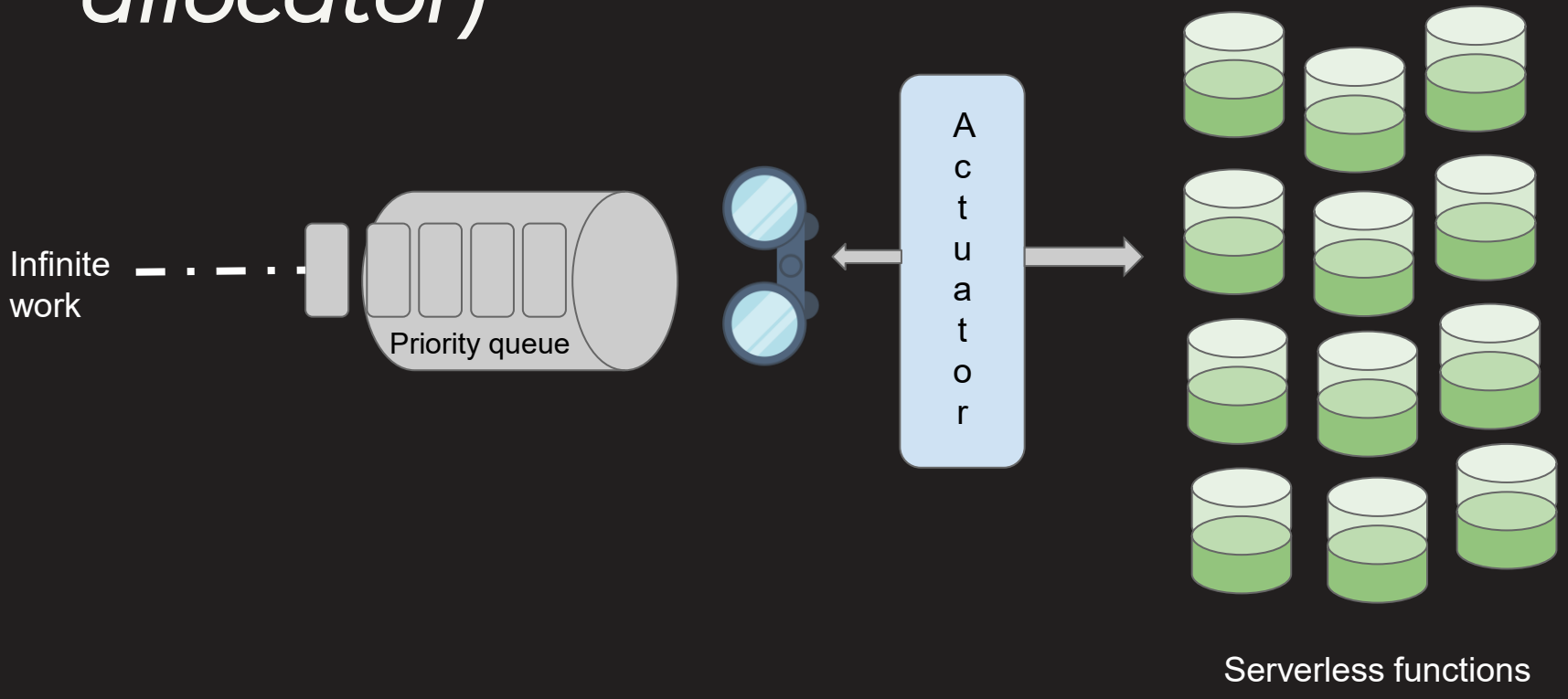


- Function's lifecycle management
- Integration with storage layers
- Initialization & execution of function code
- Resource management (CPU, memory etc.,)
- Request / response handling
- Observability tooling (logging, metrics, traces)
- Communication with messaging systems
- Security, env variables, credentials etc.,

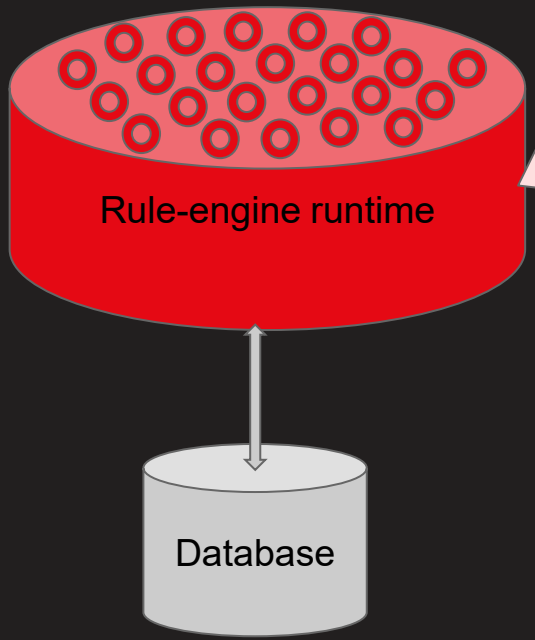
Serverless functions



Actuator (aka resource allocator)

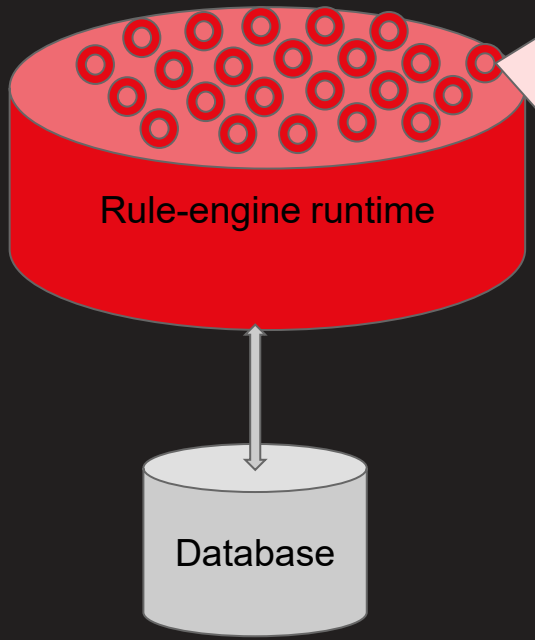


Rule-engine



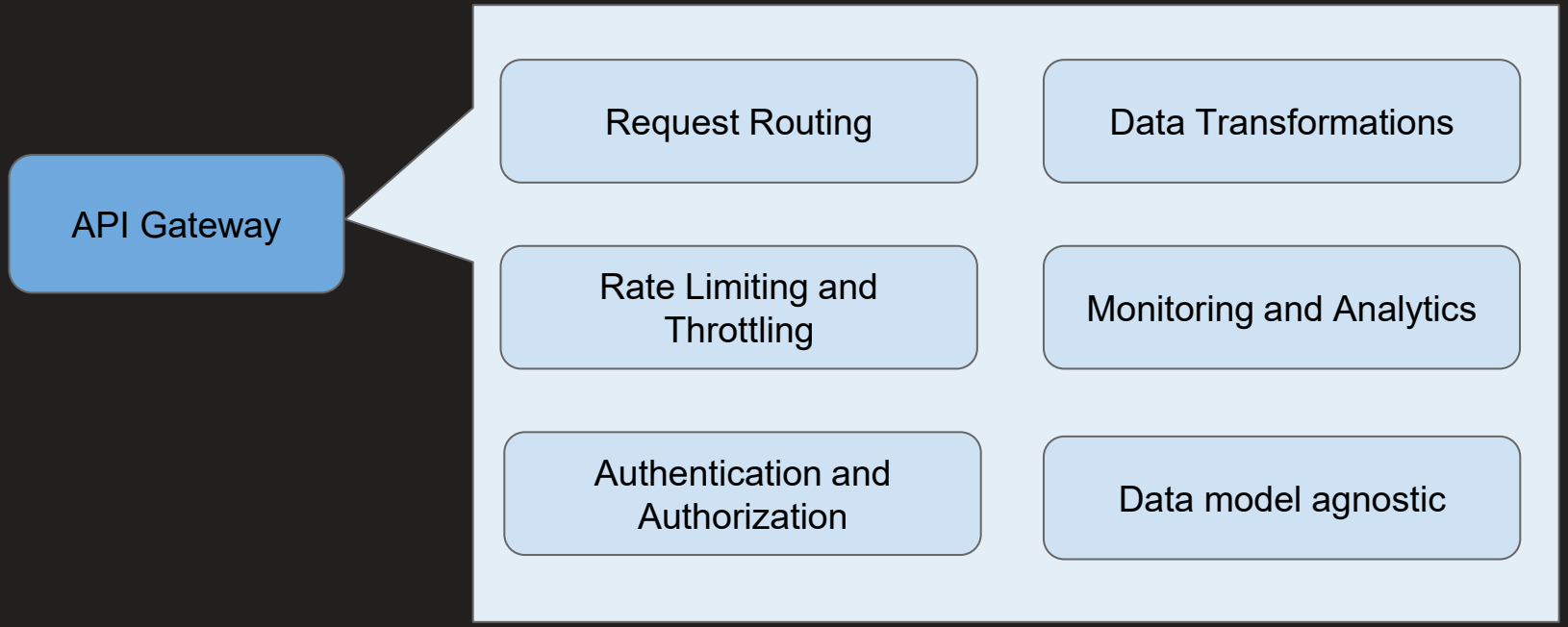
- Rule Interpreter/Processor
- Event Handling System
- Isolation Mechanism
- Security Mechanisms
- Resource Allocation and Management
- Data Interfaces
- Observability tooling (logging, metrics, traces)
- Error Handling Mechanism

Rule-engine workflows



- Rule-based workflow
- Request, notify
- Graph based workflow
- Reuse existing data
- Business logic to orchestrate multiple functions
- Leverage platform
- Error handling
- DSL based rules

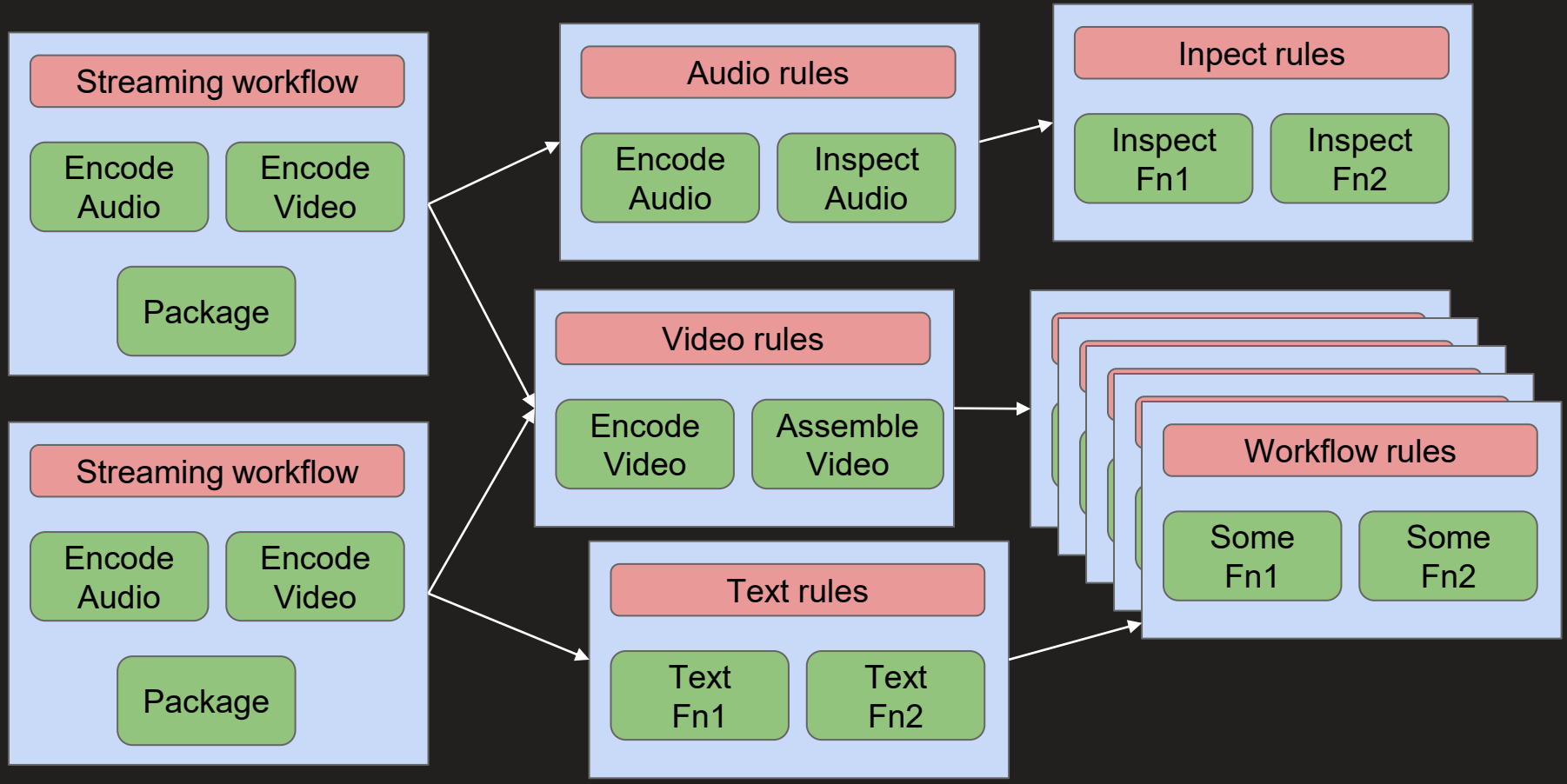
API Gateway


API Gateway
Request Routing
Data Transformations
Rate Limiting and Throttling
Monitoring and Analytics
Authentication and Authorization
Data model agnostic

Dev-Ex

- Code generator
  - Rule-based workflow
  - Graph based orchestration workflow
  - Independent function generator
- CI/CD tools
  - Config based serverless function for resource allocation
- Dashboards
  - Execution metrics
  - Cost based metrics
  - Latency metric
- UI
  - Data visibility
  - Singular place to look at anything we need

How our services look like now



Questions?

NETFLIX