Self-Documentating API-Gateway

Solozobov Evgeny, Verhovod Nikita, Polienko Artem, Kochnev Victor

Project statement

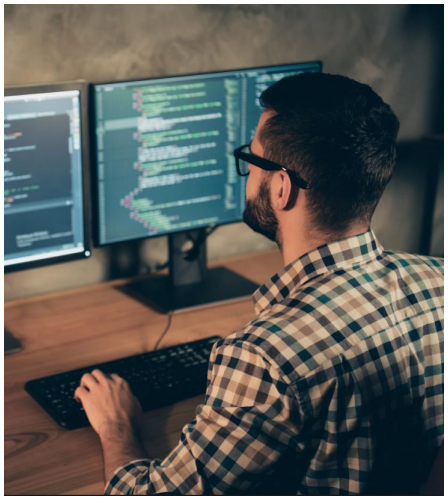
A typical ingress implementation in k8s forwards incoming traffic to a particular application. An improved API Gateway would integrate Single Sign-On, automatic OpenAPI schema generation, request validation and response caching. The goal of the project is to develop such a Kubernetes operator to extend existing ingress implementations to API Gateway. The project should provide means to define API schema using Custom Resource Definitions in K8s directly and generate them from the source code following a set of conventions.

Team: WR

Project repo: github.com/laym0n/wr

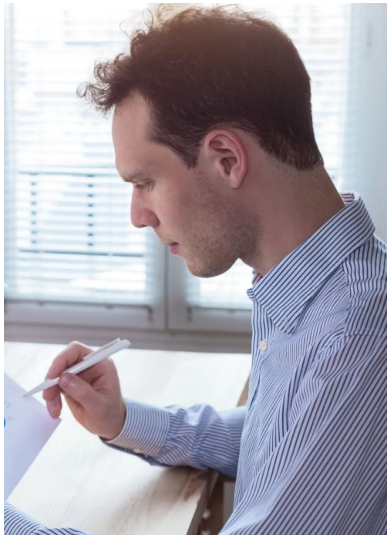
This report: tiny.cc/project-team-wr

Roles



Solution Architect

Designs the system’s architecture for scalability and availability. Uses the service mainly for API traffic management with high performance.



Developer

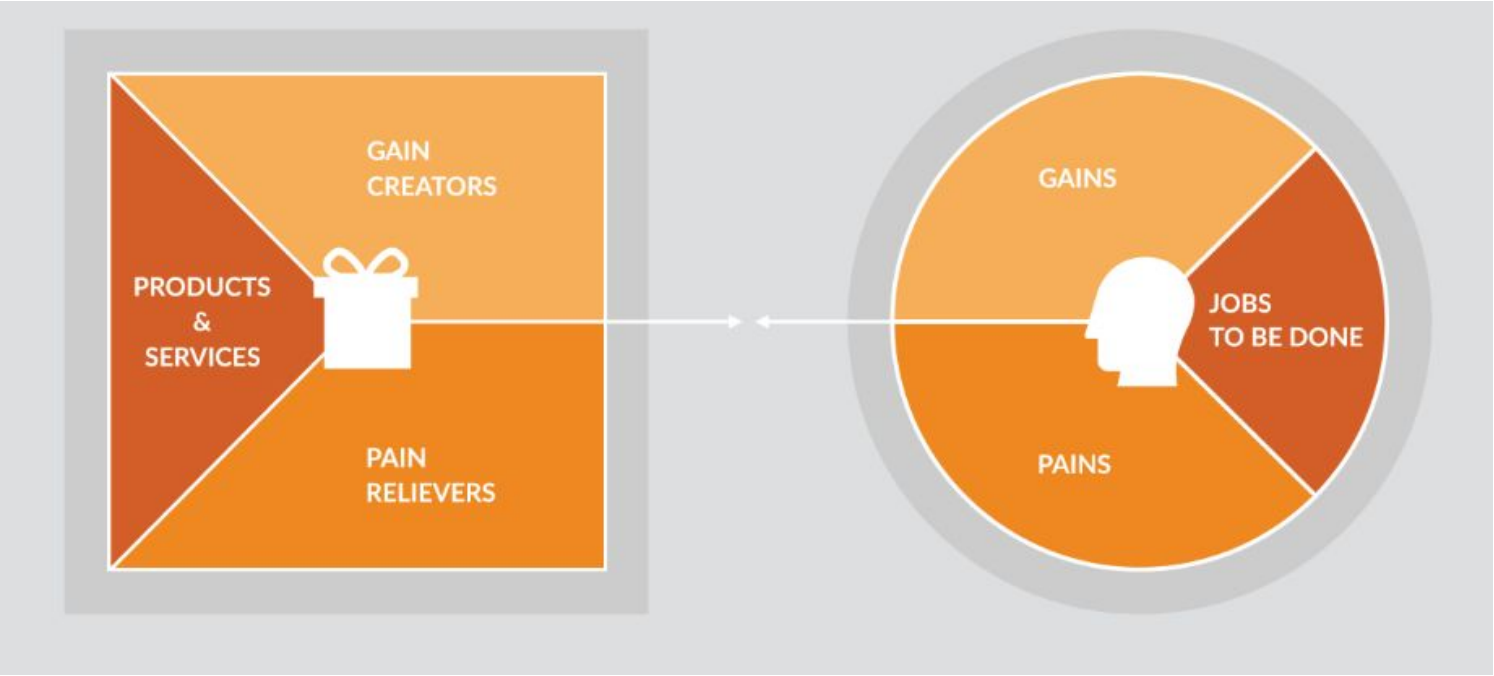
Integrates the consumer system with the API Gateway. Uses the Gateway to validate requests against the predefined API schemas.

System Analyst

Focuses on ensuring the services meets business and technical requirements. Automatic API generation reduces the time spent maintaining documentation, ensuring consistency.

Value proposition canvas (optional - for each persona)

https://drawio-app.com/create-a-value-proposition-canvas-with-draw-io/#



Jobs map (optional - for each persona)

https://drawio-app.com/wp-content/uploads/2018/10/drawio-jobs-to-be-done-white-paper.pdf

Data glossary

API Gateway: An entry point for routing, securing for clients and backend services in a Kubernetes.

Attributes: Request Validation, Response Caching, Single Sign-On (SSO), OpenAPI Schema Generation

OpenAPI Schema: A structure of an API, including its endpoints, request/response formats, and data models.

Attributes: Endpoints, HTTP Methods (GET, POST, etc.), Request Parameters, Response Structure, Data Models

Single Sign-On (SSO): A security feature that allows users to authenticate once and gain access to multiple services without needing to log in again.

Attributes: Authentication, Authorization, Token Management

Custom Resource Definition (CRD): A way to define custom API objects in Kubernetes. The CRD extends Kubernetes API functionalities by allowing custom resources specific to the API Gateway.

Attributes: Schema, Definitions, Validation Rules, API Versioning

Data glossary

Request Validation:
Ensures incoming requests adhere to the formated schema and data types in the OpenAPI specification.

Attributes: Schema Validation, Data Type Validation, Error Handling

Response Caching: Stores accessed responses to reduce backend load and improve response times.

Attributes: Cache Storage, Expiration Policy, Cache Invalidation

Operator: A Kubernetes controller that manages the lifecycle of applications as schema generator.

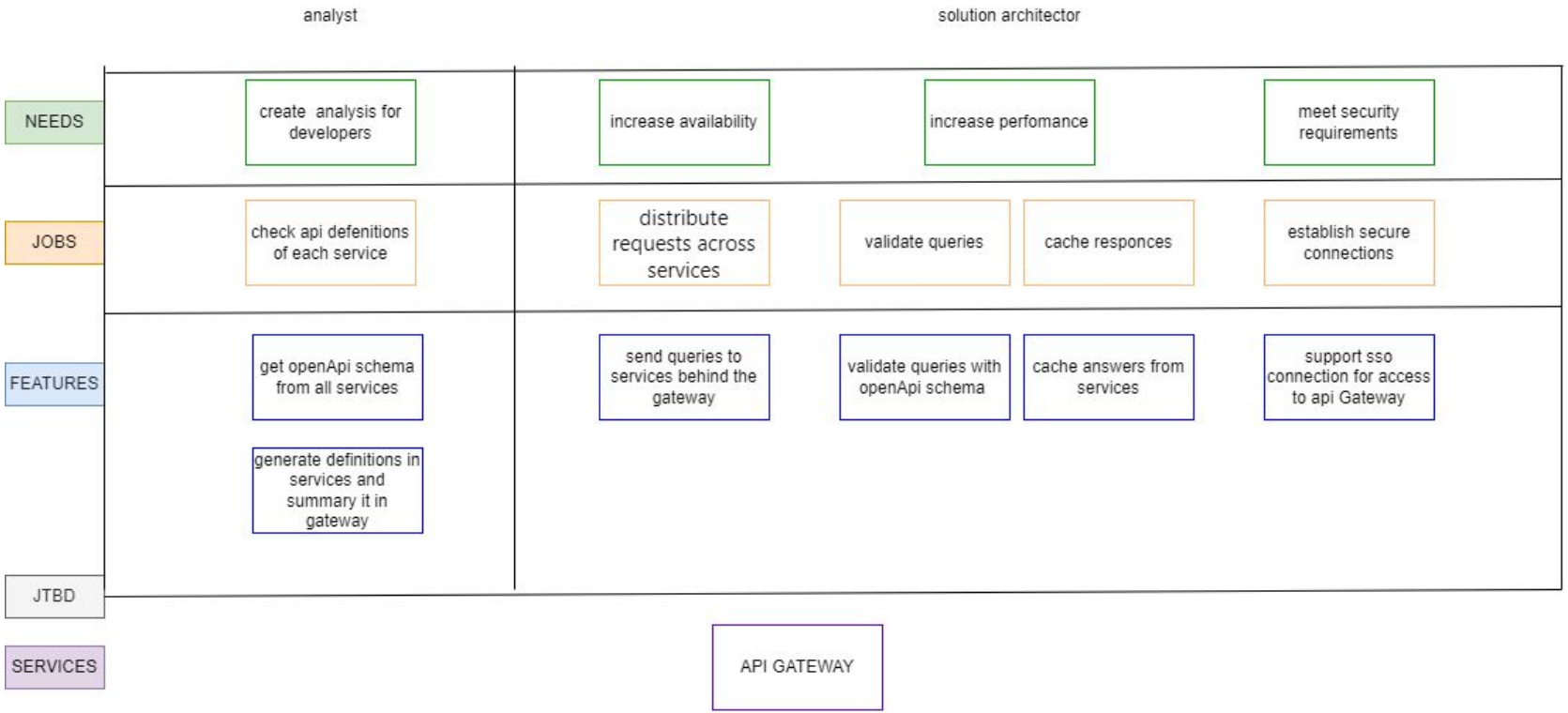
Attributes: Application Management, Event Handling, Monitoring

Data glossary

API Consumer: Any user, application, or system that interacts with the API Gateway to access backend services.

Attributes: API Key/Token, Request Frequency, Response Time

Story map


analyst
solution architector
NEEDS
JOBS
FEATURES
JTBD
SERVICES
create analysis for developers
increase availability
increase perfomance
meet security requirements
check api defenitions of each service
distribute requests across services
validate queries
cache responces
establish secure connections
get openApi schema from all services
send queries to services behind the gateway
validate queries with openApi schema
cache answers from services
support sso connection for access to api Gateway
generate definitions in services and summary it in gateway
API GATEWAY