SchwarzShuttle:

The Future of Taxi Operations

GCP Architecture for Real-Time Intelligence at Global Scale

Presented By: Thibault G

Business Context & Challenges

Our global taxi business operates at a massive scale, ensuring seamless transportation services worldwide.

❑ Operational Scale:

- Fleet of **11,000+ taxis** operating **24/7**
- Serving **millions of passengers** across multiple cities
- Integrating technology for real-time booking and tracking

❑ Key Challenges:

- **Data Management:** Handling vast amounts of trip, customer, and payment data efficiently
- **Security:** Ensuring the protection of sensitive user and financial data
- **Fraud Prevention:** Detecting and mitigating fraudulent transactions and unauthorised access
- **Optimisation:** Enhancing route planning, driver allocation, and fuel efficiency with AI-driven solutions

Solution Requirements

❑ Functional Requirements:

- Trip data collection and processing
- Real-time fleet monitoring and dispatch
- Revenue analysis and receipt generation
- Secure payment processing
- Customer communications management

❑ Non-Functional Requirements:

- **Performance:** Low latency for real-time operations
- **Scalability:** Support for global operations across time zones
- **Security:** Protection of sensitive customer and payment data
- **Reliability:** 24/7 availability with fault tolerance
- **Maintainability:** Component-level updates without service disruption

Data Ingestion & Processing Patterns

Payment Info

Fares, transaction logs, payment methods.

Processed with CMEK encryption for PCI-DSS compliance

Secured via service-specific IAM roles and data governance

Driver Ratings & Feedback

Service quality assessment.

Inputs to ML models for performance optimisation

Triggers operational alerts via Cloud Functions

Trip Data

Start/end locations, distance, duration.

Processed via Pub/Sub → Cloud Functions → BigQuery

~500MB daily data volume across fleet

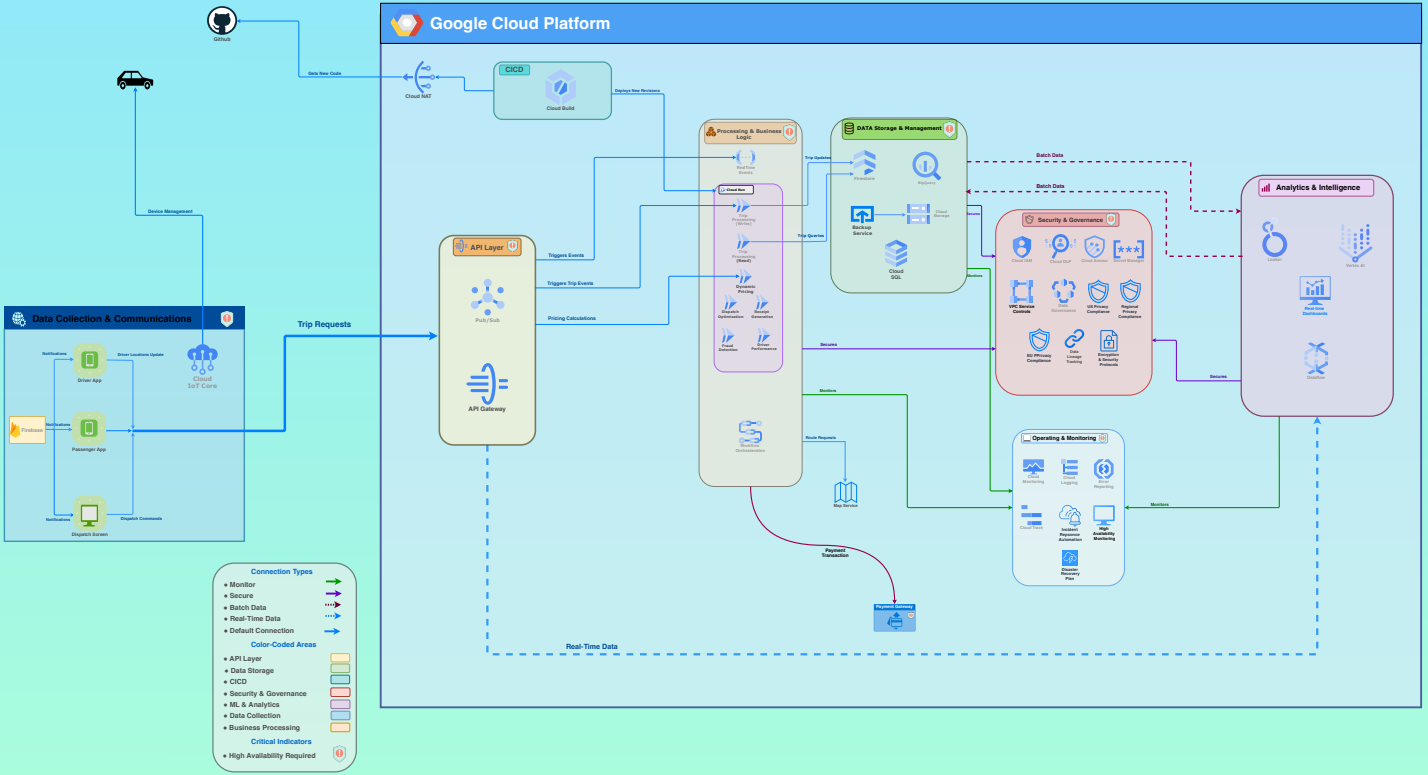
GPS Data

Real-time vehicle tracking, traffic conditions.

Streamed via Cloud IoT Core → Pub/Sub → Real-time analytics

High-frequency data (5-10 sec intervals) requiring scalable ingestion

High-Level Architecture

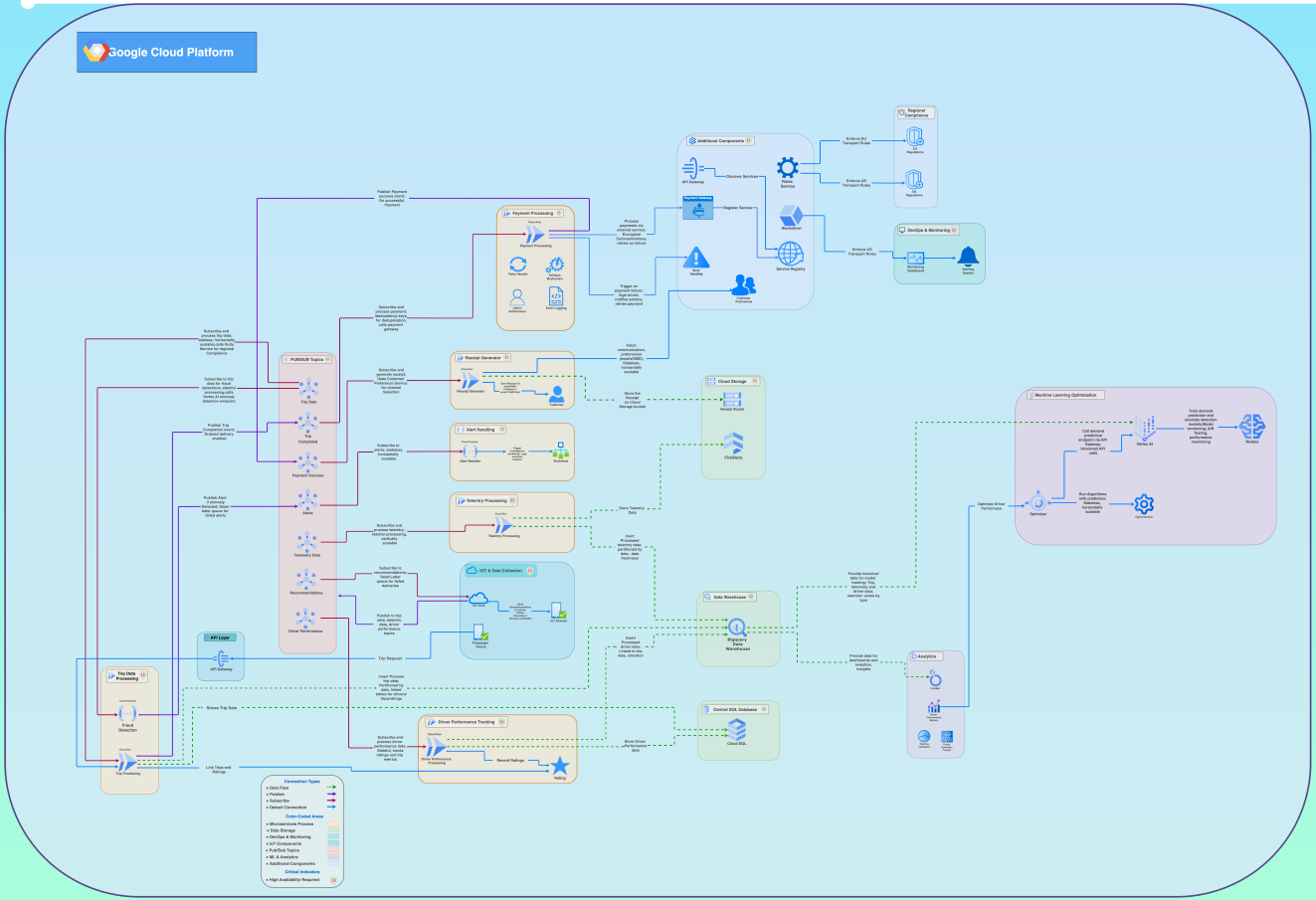


View high-resolution diagram here, and download drawio source file here

Key Points:

- **Complete Data Lifecycle:** From taxi collection to analytics insights
- **Event-Driven Design:** Real-time processing via Pub/Sub messaging
- **Layered Security:** Defense-in-depth across all components
- **API-First Approach:** Flexible integration for mobile and IoT devices
- **Intelligence Layer:** ML-powered optimization and business insights

Detailed System Architecture



View high-resolution diagram here, and download drawio source file here

Key Points:

- **Event-Driven Core:** Pub/Sub messaging enables real-time processing across all services
- **Service Specialization:** Dedicated microservices for trip, payment, and telemetry workflows
- **Data Pipeline Integration:** Seamless flow from operational systems to analytics
- **ML Feedback Loop:** Continuous intelligence from data collection to prediction deployment
- **Defense-in-Depth:** Security controls implemented at each architectural layer

GCP Implementation

Pub/Sub
(Real-Time Messaging & Event Streaming)

- *Enables **real-time ingestion** of IoT device data for fleet operations.*
- *Supports **event-driven processing**, triggering Cloud Functions for automation.*

BigQuery
(Scalable Data Warehouse)

- *Stores and analyzes trip revenue, **fleet performance**, and fraud detection data.*
- *Uses **CMEK encryption** for secure data protection.*
- *Supports **executive dashboards and analytics** for decision-making.*

Cloud Functions
(Serverless Compute for Automation)

- *Automates **data processing, security monitoring,** and fraud detection.*
- *Orchestrates workflows **triggered by Pub/Sub events**.*

Vertex AI
(Machine Learning & Optimization)

- *Powers **fleet optimization** and fraud prevention models.*
- *Enables **predictive maintenance** for vehicles.*

IAM & Security Tools
(Access Control & Threat Detection)

- *IAM enforces **least privilege access** to services.*
- *Log Sinks track security incidents in **BigQuery**.*
- *SonarQube, Trivy, OWASP ZAP integrate into **CI/CD** for security scans.*

Deployment Objectives

Security

- Protects sensitive trip and revenue data from cyber threats.
- Implements **Customer-Managed Encryption Keys (CMEK)** for secure storage.
- Enforces **IAM with least privilege access** to prevent unauthorised access.
- Integrates **log monitoring and security scans** (SonarQube, Trivy, OWASP ZAP) in CI/CD.

Scalability

- Supports **auto-scaling** to handle growing data from IoT devices.
- Ensures **low-latency** processing during peak fleet operations.
- Adapts to **multi-region** deployments for global expansion.

Optimization

- Enhances **real-time fleet management** using ML-powered insights.
- Reduces **operational costs** through intelligent resource allocation.
- Improves **user experience** with efficient and seamless data processing.

Automation Strategy

1. **Dual Approach:** Terraform (Primary) + Python (alternative)

2. **Rationale:**
   - Terraform for declarative infrastructure
   - Python for dynamic resource provisioning, complex logic

3. **Key Capabilities:**
   - Component-level deployment
   - Security configuration

4. **Benefit:** Hours → Minutes deployment time

Python Deployment Script

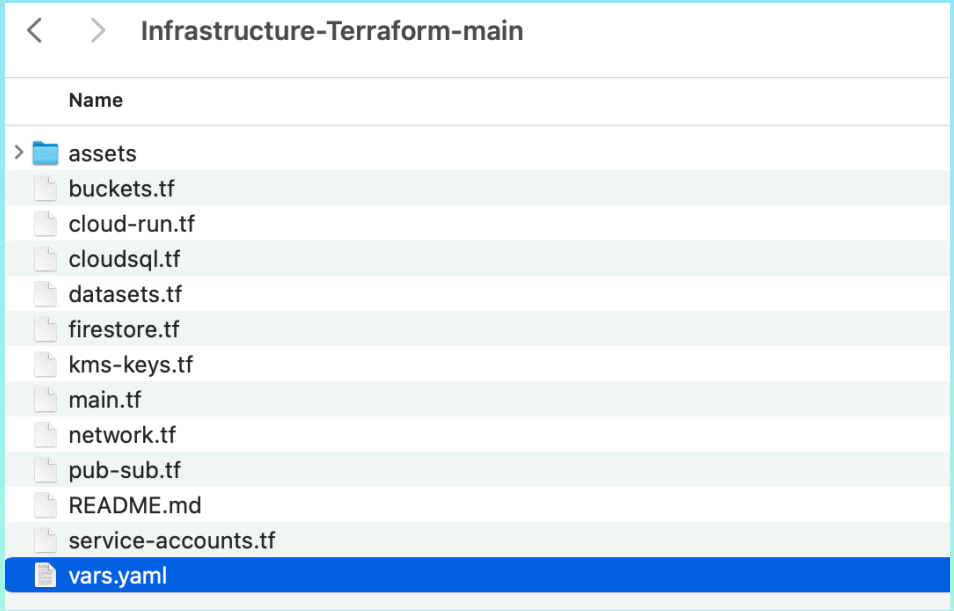
```
def create_bigquery_dataset(key_path_name):
    """Creates a BigQuery dataset with CMEK."""
    dataset = bigquery.Dataset(f"{PROJECT_ID}.{DATASET_ID}")
    dataset.default_encryption_configuration = bigquery.EncryptionConfiguration(kms_key_name=key_path_name)
    try:
        bq_client.create_dataset(dataset)
        logger.info(f"Created dataset: {DATASET_ID} with CMEK")
    except Exception as e:
        logger.warning(f"Dataset exists or error: {e}")


def deploy_cloud_functions():
    """Deploys placeholder Cloud Functions for trip processing, receipt generation, and alerts."""
    functions = [
        ("trip_processor", "trip_data-sub", "process_trips.py"),
        ("receipt_generator", "trip_completed-sub", "generate_receipts.py"),
        ("alert_handler", "alerts-sub", "handle_alerts.py")
    ]
    for name, trigger_sub, source_file in functions:
        function_path = functions_client.cloud_function_path(PROJECT_ID, LOCATION, name)
        try:
            functions_client.create_function(request={
                "location": f"projects/{PROJECT_ID}/locations/{LOCATION}",
                "function": {
                    "name": function_path,
                    "source_archive_url": f"gs://{PROJECT_ID}-functions/{source_file}",
                    "entry_point": name,
                    "event_trigger": {
                        "event_type": "google.pubsub.topic.publish",
                        "resource": pubsub_publisher.topic_path(PROJECT_ID, trigger_sub.split('-')[0])
                    },
                    "runtime": "python39"
                }
            })
            logger.info(f"Deployed Cloud Function: {name}")
        except Exception as e:
            logger.warning(f"Function {name} exists or error: {e}")
```

Browse complete repository here

Key Points:

- **Component-Level Deployment:** Each service deployed independently
- **Event-Driven Architecture:** Functions triggered by Pub/Sub events
- **Idempotent Operations**:Includes error handling for resilience
- **Configuration Flexibility:** Customisable runtime and locations
- **Audit Logging:** Deployment actions tracked for accountability

Terraform Infrastructure



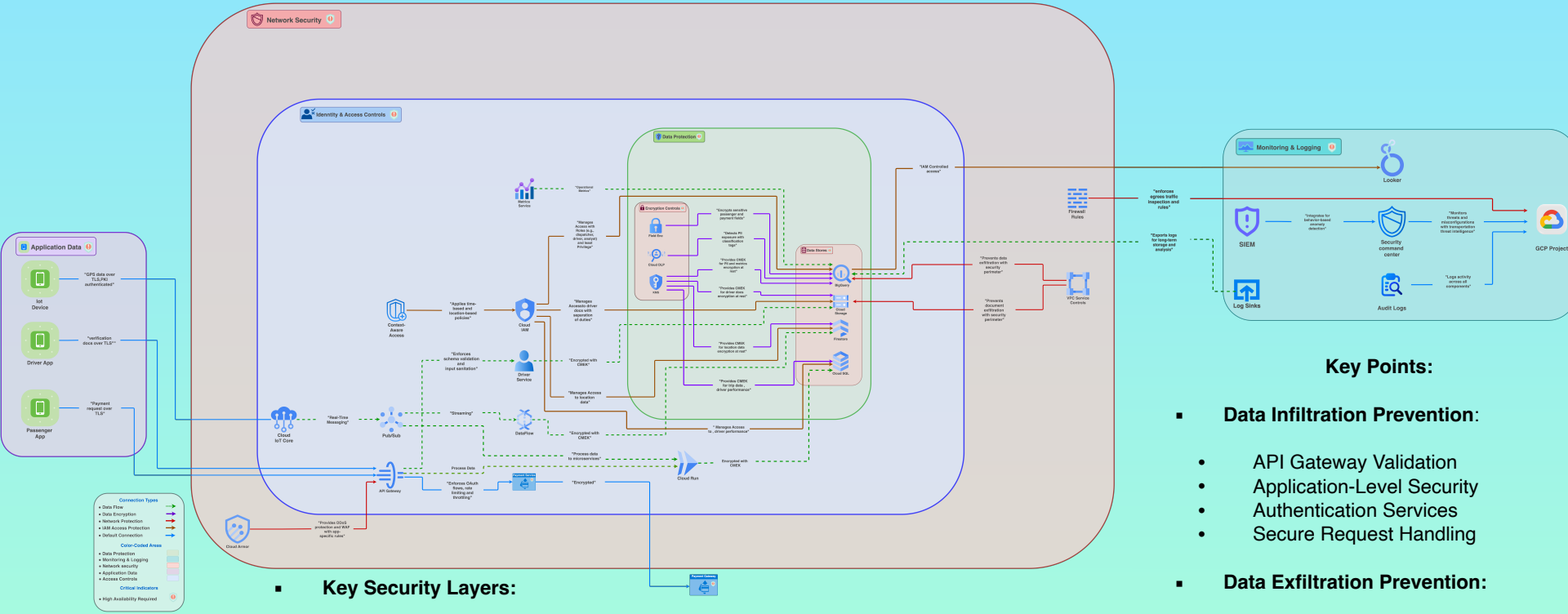

```
# Cloud Run services with secure configuration
resource "google_cloud_run_v2_service" "processing" {
  for_each            = local.yaml_vars.cloud-runs
  name                = each.value.name
  location            = local.yaml_vars.public-vars.region
  deletion_protection = each.value.deletion_protect
  ingress             = each.value.ingress

  template {
    service_account = google_service_account.service_accounts[each.value.service_account_key].email
    scaling {
      max_instance_count = each.value.max_count
    }
    containers {
      image = each.value.image
      resources {
        limits = {
          cpu    = each.value.cpu
          memory = each.value.memory
        }
      }
    }
  }
}

# Secure networking infrastructure
resource "google_compute_network" "vpc_network" {
  name                    = local.yaml_vars.network.network_name
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "private_subnet" {
  name          = "${local.yaml_vars.network.network_name}-private"
  ip_cidr_range = local.yaml_vars.network.private_subnet_cidr
  region        = local.yaml_vars.public-vars.region
  network       = google_compute_network.vpc_network.id
}
```

Key Points:

- **Configuration as Code:** Complete infrastructure defined in version control
- **Modular Architecture:** Separated services with dedicated resources
- **Security-First Design:** Private networking and service-specific accounts
- **Dynamic Scaling**: Automatic resource adjustment based on demand
- **External Configuration:** YAML-driven deployment for environment flexibility

Browse complete repository here

Multi-Layered Security & Data Protection Strategy



- **Key Security Layers:**
  - Perimeter Security: VPC, Firewalls
  - Identity & Access: IAM with Least Privilege
  - Data Protection: CMEK Encryption
  - Application Security: Input Validation, Secure APIs

- **Comprehensive Protection:**
  - Mitigating External Threats
  - Managing Insider Risks
  - End-to-End Security Approach

Key Points:

- **Data Infiltration Prevention**:
  - API Gateway Validation
  - Application-Level Security
  - Authentication Services
  - Secure Request Handling

- **Data Exfiltration Prevention:**
  - Network Perimeter Controls
  - Restricted Data Movement
  - Authorized Access Only
  - Comprehensive Logging & SIEM Monitoring

View high-resolution diagram here, and download drawio source file here

Security Implementation Controls

Python: Data Encryption & Access Control:

```
def create_kms_key():
    """Creates a KMS keyring and key for encryption."""
    location = "global"
    keyring_path = kms_client.key_ring_path(PROJECT_ID, location, KEYRING_NAME)
    try:
        kms_client.create_key_ring(
            request={"parent": f"projects/{PROJECT_ID}/locations/{location}",
                "key_ring_id": KEYRING_NAME})
        logger.info(f"Created keyring: {KEYRING_NAME}")
    except Exception as e:
        logger.warning(f"Keyring exists or error: {e}")

    key_path = kms_client.crypto_key_path(PROJECT_ID, location, KEYRING_NAME, KEY_NAME)
    try:
        kms_client.create_crypto_key(request={
            "parent": keyring_path,
            "crypto_key_id": KEY_NAME,
            "crypto_key": {"purpose": kms_v1.CryptoKey.CryptoKeyPurpose.ENCRYPT_DECRYPT}
        })
        logger.info(f"Created key: {KEY_NAME}")
    except Exception as e:
        logger.warning(f"Key exists or error: {e}")
    return key_path

def setup_security_monitoring():
    """Configures log sinks to BigQuery for monitoring."""
    # Configure audit logging
    client = gc_logging.Client(credentials=credentials)
    sink_name = "schwarzshuttle_logs"
    sink = client.sink(sink_name)

    # Filter for security-relevant logs
    sink.filter_ = f'logName:"projects/{PROJECT_ID}/logs/cloudaudit.googleapis.com%"'
    sink.destination = f"bigquery.googleapis.com/projects/{PROJECT_ID}/datasets/{DATASET_ID}/tables/audit_logs"

    sink.create()
    logger.info(f"Created Log Sink '{sink_name}' to BigQuery.")
```

Terraform: Network & Security Configuration

```
# Secure service identity management
resource "google_service_account" "service_accounts" {
  for_each     = local.yaml_vars.service-accounts
  account_id   = each.value.name
  display_name = each.value.display_name
}

# Customer-managed encryption keys
resource "google_kms_key_ring" "key_ring" {
  name     = local.yaml_vars.kms-keys.keyring.name
  location = local.yaml_vars.public-vars.region
}

resource "google_kms_crypto_key" "crypto_key" {
  for_each = local.yaml_vars.kms-keys
  name     = each.value.name
  key_ring = google_kms_key_ring.key_ring.id
}

# Encrypted data storage with fine-grained access control
resource "google_bigquery_dataset" "dataset-test" {
  for_each   = local.yaml_vars.Datasets
  dataset_id = each.value.dataset_id
  location   = local.yaml_vars.public-vars.region
  depends_on = [google_kms_crypto_key.crypto_key]

  default_encryption_configuration {
    kms_key_name = local.yaml_vars.kms-keys.dataset-key.name
  }
}

# Role-based access control for data
resource "google_bigquery_dataset_iam_binding" "data-reader" {
  for_each   = local.yaml_vars.Datasets
  dataset_id = google_bigquery_dataset.dataset-test[each.key].dataset_id
  role       = "roles/bigquery.dataViewer"
  members    = each.value.reader_members
}
```

Key Security mechanisms demonstrated:

- **Customer-Managed Encryption Keys**
- **Least Privilege Access Control**
- **Fine-Grained IAM Bindings**
- **Security Monitoring**
- **Resource Isolation**

Security Enhancement

Threat Model for Taxi Operations

- **Passenger Data**: Personal information, trip history
- **Payment Security**: Transaction processing, fraud prevention
- **Location Privacy**: Trip tracking, route protection

GCP Security Implementation

- **Identity:** Cloud Identity MFA for all access
- **Encryption:** CMEK for data-at-rest & in-transit
- **Access Control:** API Gateway with Cloud Armor
- **Network Isolation**: VPC Service Controls

Regulatory Compliance

- **PCI-DSS:** Payment card protection
- **GDPR/CCPA**: Privacy controls
- **Industry:** Required ride record retention

GCP Defense-in-Depth for Taxi Operation

Monitoring & Incident Response

Security Command Center
Cloud Logging
Event Threat Detection

Endpoint & Identity Security

Cloud Identity
IAM service accounts
Conditional access

Data Security

CMEK encryption
Cloud KMS
Cloud DLP

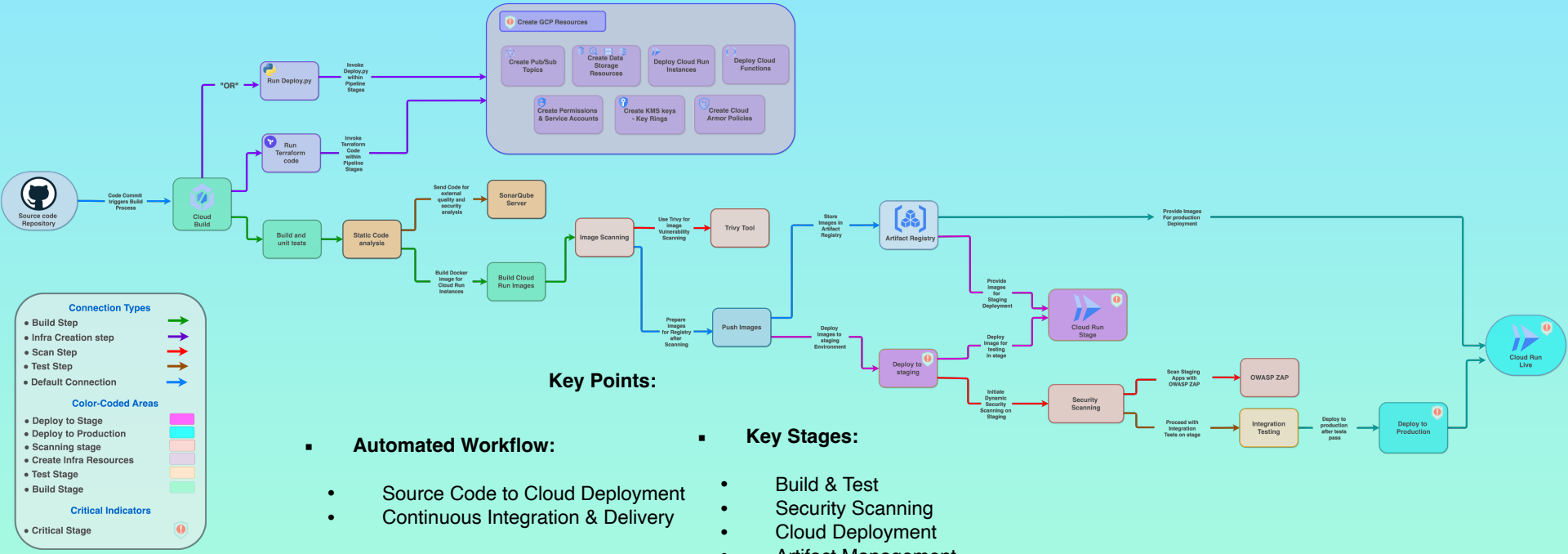
Application Security

Cloud Armor
API Gateway
Container security

Perimeter Security (Network Level)

VPC Service Controls
Cloud NAT
Secure IoT connectivity

DevSecOps Pipeline



View high-resolution diagram here, and download drawio source file here

Key Points:

- **Automated Workflow:**
  - Source Code to Cloud Deployment
  - Continuous Integration & Delivery
- **Security Integration:**
  - Image Scanning
  - OWASP ZAP Security Checks
  - Vulnerability Assessment
- **Key Stages:**
  - Build & Test
  - Security Scanning
  - Cloud Deployment
  - Artifact Management
- **Advanced Features:**
  - Cloud Resource Provisioning
  - Automated Security Policies
  - Multi-Environment Support

Measurable Improvements

Operational Metrics:

- 30% reduction in passenger wait times
- 15% decrease in vehicle idle time
- 8% increase in revenue per mile

Technical Metrics:

- 50% faster deployment time
- 60% reduction in configuration errors
- 25% reduction in security incidents
- 30% less service disruption during updates

Our Journey: From Launch to Leadership (2025-2027)

Year 1

- **Enhanced Analytics**
- **Infrastructure Improvements**
- **Security Advancements**

Year 2

- **Predictive Intelligence**
- **Customer Experience**
- **Operational Efficiency**

Year 3

- **Autonomous Capabilities**
- **Sustainability Focus**
- **Advanced Analytics**

Expected Business Outcomes:

- **Higher utilization rates** (less idle time).
- **Better customer retention** (faster rides, optimized pricing).
- **Cost reduction** (fuel savings, route efficiency).

Requirements Fulfilled

Requirement	Implementation	Benefits
Architecture	Complete GCP solution with real-time processing and bidirectional communication	Comprehensive data management, operational control, actionable insights
Automation	Terraform (primary) + Python script (alternative) with component-level updates	Consistent deployments, operational flexibility, faster updates
Security	Defense-in-depth with CMEK, IAM, monitoring	Protected customer data, regulatory compliance, threat prevention
Vision	Strategic 3-year roadmap (2025-2027)	Clear evolution path aligned with business growth

Key Takeaways

- **Comprehensive Architecture** – End-to-end solution for global taxi operations.
- **Flexible Automation** – Terraform + Python options for deployment preferences.
- **Security By Design** – Multi-layered protection for sensitive data.
- **Measurable Impact** – Clear improvements in key business metrics.
- **Strategic Vision** – Forward-looking roadmap for continued innovation.

Driving the Future of Intelligent Mobility

From Data to Movement, From Insights to Action