



gable

**Gable: The Easy Way
to Do for Data What
DevOps Did for Code**

Table Of Content

Chapter 1: The Problem 03

- Code has DevOps
- Data doesn't
- Engineers find out too late
- Data quality issues aren't just engineering problems

Chapter 2: What Gable Does 04

- Detects data-impacting changes in application code
- Enforces data contracts as code
- Validates changes pre-deployment
- Automated notifications & impact analysis
- Governance & Trust at the Code Level

Chapter 3: How Gable Works 05

- Scans source code & data logic before deployment
- Enforces contracts at the source
- Blocks breaking changes in CI/CD pipelines
- Notifies the right teams proactively
- Tracks version-controlled migrations & rollbacks

Chapter 4: Example: How Gable Prevents Costly Mistakes 06

- Without Gable
- With Gable

Chapter 5: Why Teams Love Gable 07

- Prevents bad data before it reaches production
- Ensures smooth team collaboration
- Ensures data is trustworthy for business decisions



Chapter 1: The Problem

- ✓ Code has DevOps—automated testing, static analysis, and CI/CD ensure reliability before deployment.
- ✓ Data doesn't. Schema changes, transformations, and serialization updates often break downstream systems without warning.
- ✓ Engineers find out too late—only after broken dashboards, failed ML models, or outages disrupt the business.
- ✓ Data quality issues aren't just engineering problems—they're business problems.
 - Inconsistent data causes inaccurate reporting.
 - ML models train on corrupted or misaligned data.
 - Executives lose trust in analytics and ignore data-driven insights.



Chapter 2: What Gable Does

Gable applies DevOps-style static analysis to data, preventing failures before deployment and ensuring data remains trustworthy, accurate, and aligned across teams.



Detects data-impacting changes in application code → Scans schemas, transformations, and data structures before they go live.



Enforces data contracts as code → Codifies agreements between teams to ensure reliable, consistent data.



Validates changes pre-deployment → Catches schema drift, type mismatches, missing fields, and unexpected null values before they cause business problems.



Automated notifications & impact analysis → Notifies relevant teams of upcoming changes so they can adapt before production is affected.



Governance & Trust at the Code Level → Tracks who changed what, why, and its downstream impact, ensuring executives can trust data-driven decisions.



Chapter 3: How Gable Works

- ✓ Scans source code & data logic before deployment → Just like static code analysis for software, Gable prevents bad data from shipping.
- ✓ Enforces contracts at the source → Ensures producers follow agreed data structures, types, and policies.
- ✓ Blocks breaking changes in CI/CD pipelines → If a proposed change violates a contract or impacts a critical business metric, Gable flags or blocks it.
- ✓ Notifies the right teams proactively → Engineers, analysts, and business stakeholders get early visibility into data changes so they can adjust before mistakes happen.
- ✓ Tracks version-controlled migrations & rollbacks → Supports automated recovery when changes introduce errors.



Chapter 4: Example: How Gable Prevents Costly Mistakes

Without Gable

A backend team removes a field (`customer_id`) from an API response. The data engineering team isn't notified. Pipelines break, and reports show incorrect revenue. Executives make strategic decisions based on bad data. Trust in data erodes—teams start questioning whether they can rely on analytics at all.

With Gable

- ✓ Gable detects the schema change in code before deployment.
- ✓ It alerts both engineering and analytics teams that a breaking change would impact key revenue reporting.
- ✓ The teams collaborate upfront to ensure data integrity before rolling out the change.
- ✓ No outages, no broken dashboards, no bad business decisions.



Chapter 5: Why Teams Love Gable

Gable applies DevOps-style static analysis to data, preventing failures before deployment and ensuring data remains trustworthy, accurate, and aligned across teams.



Prevents bad data before it reaches production → Proactive, not reactive.



Ensures smooth team collaboration → Data producers and consumers stay in sync, avoiding confusion and misalignment.



Ensures data is trustworthy for business decisions → Executives can rely on accurate, high-quality analytics.



Real-World Impact

"Gable has helped us bring trust to petabyte-scale data by ensuring governance at the source. We no longer spend engineering cycles fixing broken data downstream." — Glassdoor Engineering Team



Learn More: Download "[How Gable Works, A Technical Breakdown](#)"