



How Gable Works: A Technical Breakdown

Table Of Content

Chapter 1: Core Functionality: What Gable Does 04

- Static Code Analysis for Data Changes
- Detects Non-Schema Breaking but Data-Impacting Changes
- Enforces Data Contracts at the Code Level
- Real-Time Impact Analysis & Team Notifications
- CI/CD Integration for Pre-Deployment Validation
- Continuous Version Control for Data Code & Migrations

Chapter 2: Why Gable is Different 08

- Prevents Data Issues at the Code Level
- Developer-First, Integrated with CI/CD
- Code-Driven Policy Enforcement

Summary 10

Gable is **not** a data observability tool. It's a **shift-left data governance platform** that enforces **data contracts and proactive monitoring at the application code level** before data issues impact downstream systems.

Unlike traditional tools that focus on **database schema changes**, Gable detects **application-level changes in data logic, transformations, and structures**—preventing silent data integrity issues **before they reach storage, pipelines, or analytics tools**.





Chapter 1: Core Functionality: What Gable Does

Static Code Analysis for Data Changes

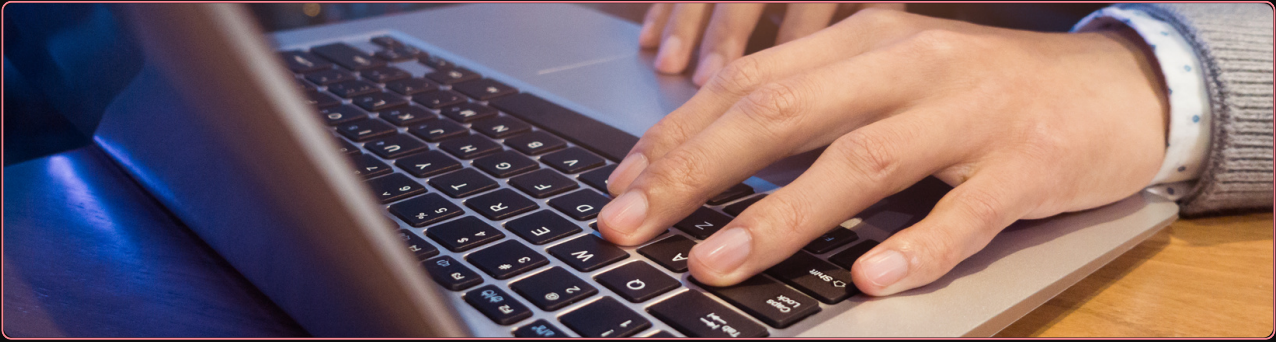
- ✓ Analyzes source code to detect modifications in data structures, variable types, serialization logic, and business rules.
- ✓ Prevents hidden data integrity issues caused by application logic, not just schema changes.
- ✓ Example: If a function changes how timestamps are processed (e.g., from local time to UTC), Gable flags potential impacts on data accuracy before deployment.

Detects Non-Schema Breaking but Data-Impacting Changes

- ✓ Captures code-level transformations that affect how data is structured, interpreted, and consumed.
- ✓ Tracks changes such as:
 - Serialization format modifications (e.g., switching from JSON to Protobuf).
 - Implicit type casting (e.g., converting int to float can impact financial precision).
 - Null handling and default values (e.g., replacing NULL values with a default that misrepresents data).
- ✓ Example: An API response format changes from {user_id, name} to {id, username}, potentially breaking downstream systems relying on user_id mappings.

Enforces Data Contracts at the Code Level

- ✓ Defines and validates expected data structures, formats, and transformations directly in code.
- ✓ Prevents breaking changes before they impact production by ensuring contracts are upheld across teams.
- ✓ Blocks risky deployments or triggers approval workflows if a data contract violation is detected.
- ✓ Example: If an internal API changes how transactions are structured, Gable ensures all consuming services are aware and adapted before deployment.



Real-Time Impact Analysis & Team Notifications

- ✓ Automatically maps dependencies between data producers and consumers.
- ✓ Notifies affected teams before a breaking change propagates.
- ✓ Example: If a backend team modifies a key event structure in Kafka, Gable alerts downstream ML pipelines and BI teams that rely on this data.

CI/CD Integration for Pre-Deployment Validation

- ✓ Hooks into GitHub, GitLab, Jenkins, and other CI/CD tools to validate code-level data changes before merge/deployment.
- ✓ Uses pre-deployment checks to ensure all structural and behavioral changes comply with predefined data contracts.
- ✓ Example: If a change alters how an application processes financial transactions, Gable blocks the deployment if it risks introducing rounding errors.

Continuous Version Control for Data Code & Migrations

- ✓ Tracks all changes to data logic and transformations as versioned artifacts.
- ✓ Supports automated rollback of incorrect data changes to minimize production impact.
- ✓ Example: A faulty data processing step in a data producing application can be automatically reverted to a previous working version.



Chapter 2: Why Gable is Different

Prevents Data Issues at the Code Level

- ✓ Unlike observability tools that detect issues after they occur, Gable prevents data failures before they happen.
- ✓ Unlike schema-focused tools, Gable detects logic changes in code that impact data integrity.

Developer-First, Integrated with CI/CD

- ✓ Works within existing engineering workflows (GitHub, Jenkins, dbt, Databricks).
- ✓ Prevents bad data changes before they reach pipelines, dashboards, or ML models.

Code-Driven Policy Enforcement

- ✓ Data contracts and validation rules are enforced in code, ensuring data reliability is a standard development practice.
- ✓ Supports automated testing for data transformations to catch errors before they reach production.



Summary

Feature	What Gable Does
Static Code Analysis for Data Changes	Detects data structure & logic modifications in source code before deployment.
Automated Enforcement of Data Contracts	Ensures expected formats, types, and transformations remain stable.
Impact Analysis & Team Notifications	Identifies affected consumers (dashboards, ML models, ETL pipelines) before deployment.
CI/CD Integration & Deployment Guardrails	Blocks risky changes and enforces validation in code reviews.
Version-Controlled Migrations & Rollbacks	Tracks changes in data transformations & supports automated recovery.
Infrastructure as Code for Data Governance	Applies policy-driven governance through code-driven validation.



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 Glassdoor Built Data Trust at Petabyte Scale with Gable](#)"