

# Creating a Golden Path for GraphQL

Making Success the Path of Least Resistance

**Benjie Gillam & Kewei Qu**

GraphQL Technical Steering Committee

# Who is the GraphQL TSC ?

## GraphQL Technical Steering Committee (TSC)

Consists of 11 members driving the technical direction of the GraphQL ecosystem.

### Governance & Leadership

Includes **Lee Byron**, Executive Director of the GraphQL Foundation.

Composed of **10 elected members** serving two-year terms.

### Industry Representation

Members represent major players: **Meta, Netflix, Yelp, 1stDibs**, and key open source projects.

## Scaling Success for Everyone

We work closely with organizations from the **smallest startups** to the **largest enterprises**.

# GraphQL, we have a problem

## The Pattern of Adoption

We actively monitor feedback across GitHub, social media, blogs, and news. A common, concerning pattern emerges: **adopters start with high enthusiasm, but ultimately give up.**

## The Frustrating Truth

For successful practitioners, these post-mortems are frustrating to read. **The solutions to these exact problems have existed since GraphQL was open-sourced in 2015.**

*"This is not user error — it is a failure to lead users to the pit of success."*

# The Pit of Success vs. The Pit of Despair



## The Status Quo

Companies with a **GraphQL champion** succeed far more often, but not every organization has access to such expertise.



## The Goal

**Ensure success from the very outset, not after the fact.**

Guide users naturally into the "**Pit of Success**".

# Introducing the **Golden Path** Initiative

*The default delightful experience for users.*



**Identify Common Problems**



**Outline Standard Solutions**



**Encourage Ecosystem Adoption**



**Minimize Downsides**

# Golden Path: Guiding Principles



Innovation is Welcome



Consistency & Documentation



Healthier Ecosystem



Community Contributions





## The Path is Still Being Built

We are in the building phase. Your input, experiences, and feedback matter as we define these recommendations!




# Common Problem #1 — The N+1 Problem

One of the most frequent and impactful GraphQL performance complaints.

## The Amplification

-  Nested execution model can **amplify the N+1 problem significantly** compared to traditional REST APIs.
-  Without intervention, resolving nested lists results in an exponential number of database or service calls.

## The Standard Solution & Its Flaws

-  **DataLoader** (launched alongside GraphQL) is the standard solution for batching and caching.
-  However, DataLoader must be **known about** and **used consistently everywhere** in the schema.
-  Inconsistent use leads to small batch sizes, degraded performance, and ultimately **production downtime**.

# Proposed Solution: **Batched Execution**

Batched execution addresses N+1 at the framework level by default.

## **Forms**

Simple batch resolvers

## **The Goal**

Every call to your business logic should be **batched automatically**.

## **The Impact**

N+1 becomes a **solved problem**, not a late-night incident.

## Common Problem #2 — Poor Fragment Usage

---

### ✘ The Misuse

---

- ⚠ Fragments are often completely **unused** by developers new to GraphQL.
- </> When used, they are frequently misused purely as a way to **DRY up code** (Don't Repeat Yourself).
- ✂ Treating fragments merely as text substitution misses their true architectural power.

### ✔ The Intended Design

---

- 🧩 Fragments were designed for **componentizing data needs**, not just reuse.
  - 📄 **Correct use:** Co-locate data requirements directly with the component definition.
-

# More Topics & Your Input



Cost Limits & Pagination Limits



Error Masking & Security Defaults



Naming Conventions & Schema Design



Native Pagination Support



## Share Your Experience

- What problems did you face when adopting GraphQL?
- What solutions worked — or didn't — and why?



## Take Action

- Submit your thoughts via an issue on the [Golden Path WG repository](#).
- Join a [working group meeting](#) — they're free to attend!

# GraphQL Is Now a Mature Technology

---

# 10

# YEARS

of production usage & learnings



It is time to reflect and **encode best practices** into the default experience.



The Golden Path turns a decade of hard-won knowledge into **accessible defaults**.



A huge, **coordinated community effort** is needed to make this a reality.



**Successful practitioners, library maintainers, and doc writers: join us!**