



Goodbye AWS Lambda, Hello DBOS

October 23, 2024

Agenda

- Intro to DBOS
- Intro to TMG Fuse
- TMG journey with DBOS
 - Why Serverless?
 - Building Fuse on DBOS vs. AWS Lambda
- Developing with DBOS
 - Demo
- Q&A

DBOS founded on 3 years of MIT-Stanford joint R&D



Mike Stonebraker
DBOS CTO
MIT, Turing award winner,
Creator of Postgres



Matei Zaharia
Founding Advisor
Creator of Apache Spark
Databricks CTO

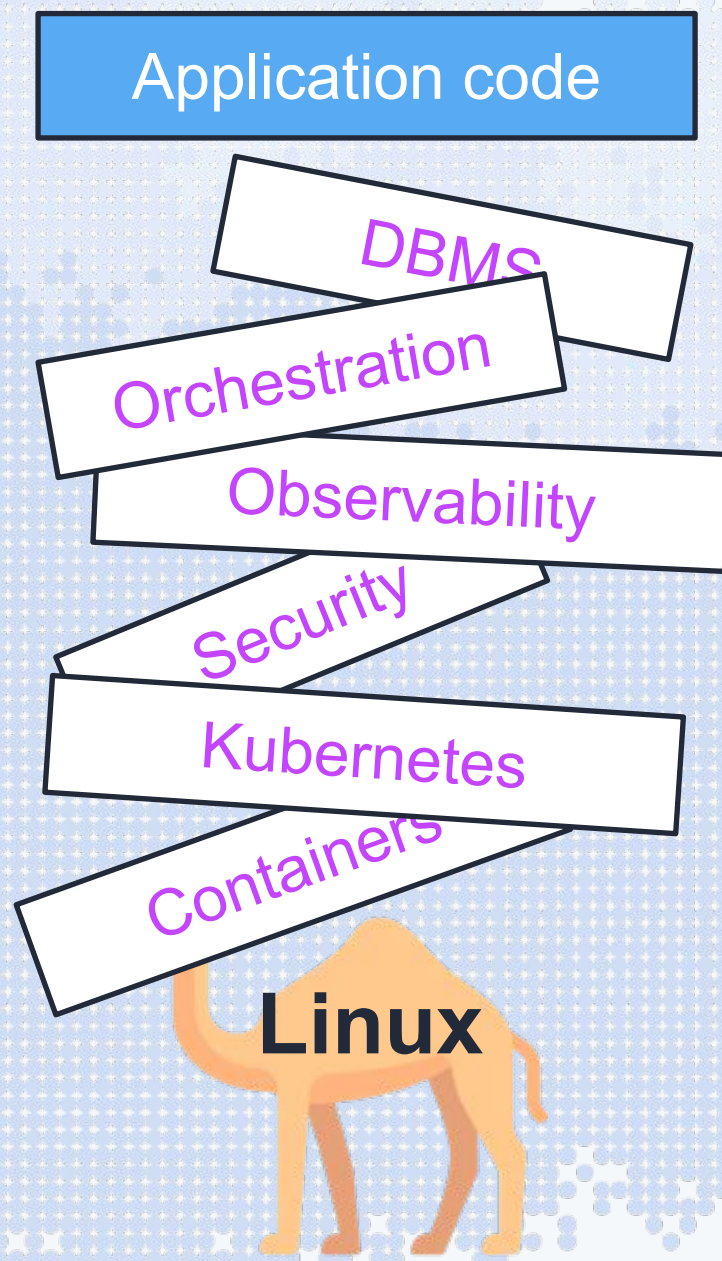


Qian Li
Co-founder/Architect
Stanford PhD



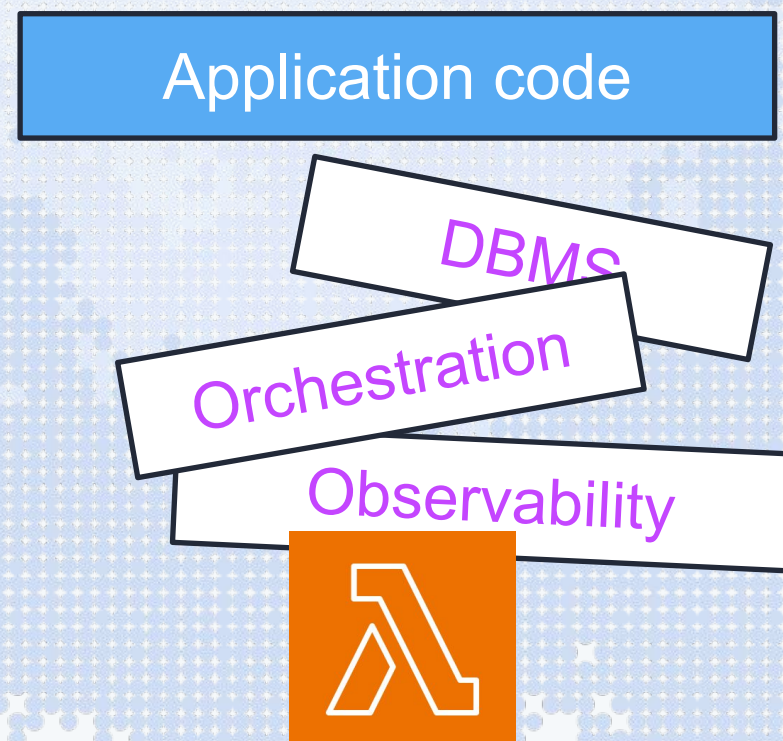
Peter Kraft
Co-founder/Architect
Stanford PhD

Founders' Observation: Cloud Development is Broken



Traditional cloud-native stack

- Built on a 35-year-old OS
- Too many moving parts
- Too many admins
- Too many security attack points
- OS can't scale to store state
- Too much hosting cost



Serverless stack

- Much simpler, but...
- “A spiderweb” at scale
- C-I-Y statefulness, observability, durability
- Pay for wait time – makes data- and LLM-driven backends costly

The DBOS Paradigm Shift

Your Applications

DBOS

Postgres
Database

Firecracker
VMs

AWS Bare Metal

Stateful serverless

- Execution platform built on a DBMS
- Crashproof, observable, stateful by default
 - Less coding!
 - No need for AWS Step Functions
- Faster performance
- No paying for “wait time”

TMG is a full-service digital experience & engineering firm.

We specialize in complex B2B solutions. We unravel the many difficulties in commerce, design, experience, and strategy that evolving businesses face. Our mix of B2B expertise, agile methodology, and approachable, person-to-person partnership delivers quicker time to value and a dynamic, informed strategy for success.

Established

2014

Global Employees

250+

Preferred Partners:



Focus Areas



Agile

Continuous delivery keeps brands moving. Continuous testing keeps brands ahead of the competition. As experts in agile work, our teams deliver value and innovation quickly.



Strategy

Business agility begins with strategy. To achieve digital transformation, your strategy has to encompass all internal processes and governance that are needed to support business agility.



Experience

Our approach centers around rapid and transparent communication, dovetailing research with design, and applying time-tested methodology to create full-phase customer experiences and meaningful, measurable moments.



Technology

We apply tailored technical expertise, practical experience, and deep platform knowledge to deliver industry-best solutions so our customers can achieve fast time-to-value. Our work is focused on scalable, future-proof technology empowering businesses beyond tomorrow.

Meet the TMG Team



Thomas McNally
VP of Technology
TMG



Aaron Brill
Commerce Engineering Lead
TMG



TMG Fuse for Shopify

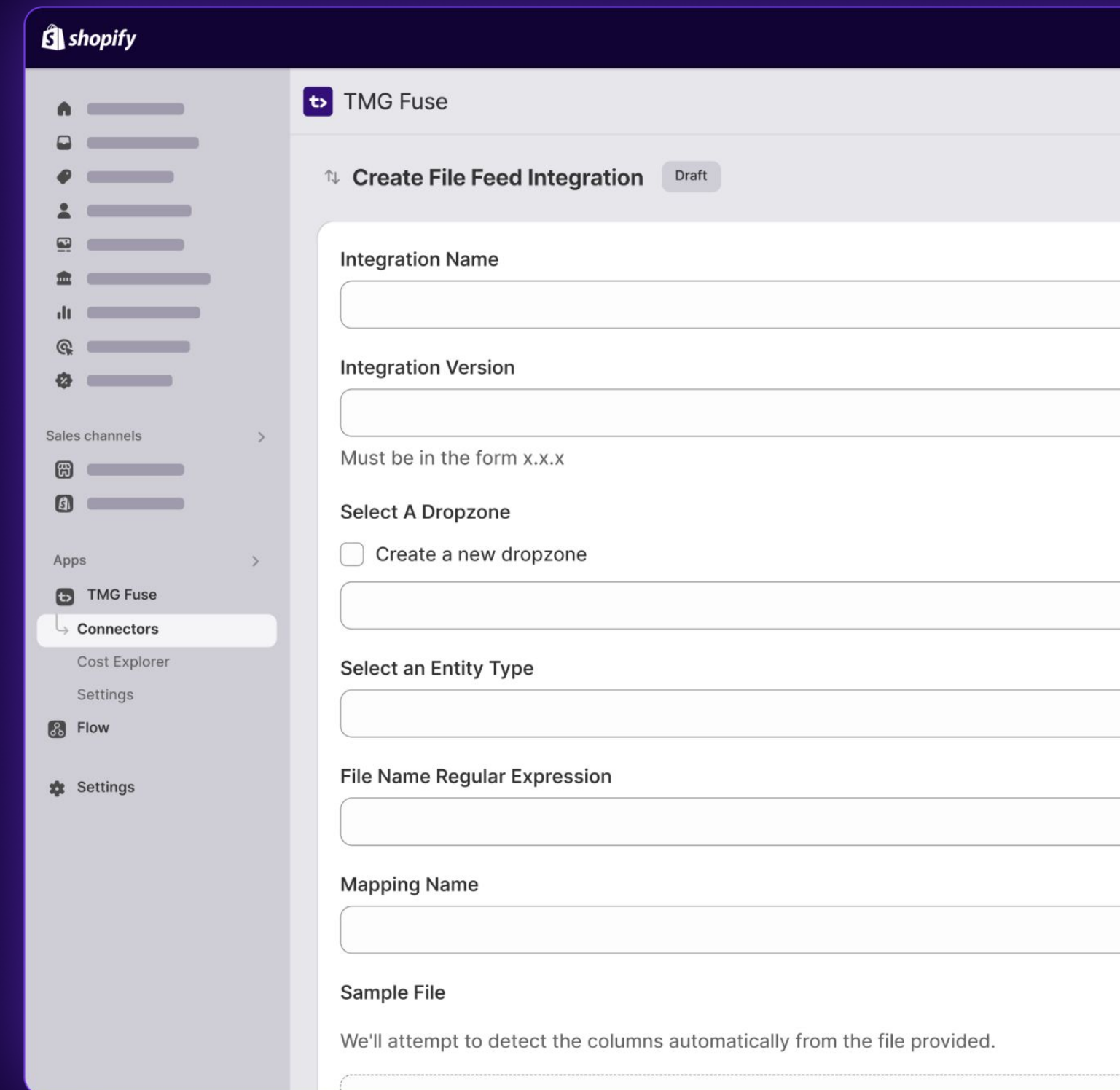
Integrate Once, Surface Data Everywhere

THE BASICS

TMG Fuse for Shopify

Introducing TMG Fuse, an enterprise-ready integration platform designed to facilitate seamless data flow between your Shopify stores and your external systems like ERP and CRM.

Fuse is a comprehensive solution for creating, connecting, managing, and monitoring integrations, tailored for tech-savvy merchants, business teams, and developers.



The screenshot displays the TMG Fuse interface within a Shopify store. On the left is a sidebar menu with icons for home, notifications, orders, customers, products, analytics, and settings. Below these are sections for 'Sales channels', 'Apps' (including TMG Fuse), 'Connectors', 'Cost Explorer', 'Settings', 'Flow', and 'Settings'. The main content area is titled 'TMG Fuse' and features a 'Create File Feed Integration' button with a 'Draft' status. The form includes several input fields: 'Integration Name', 'Integration Version' (with a note 'Must be in the form x.x.x'), 'Select A Dropzone' (with a checkbox for 'Create a new dropzone'), 'Select an Entity Type', 'File Name Regular Expression', 'Mapping Name', and 'Sample File'. A note at the bottom states, 'We'll attempt to detect the columns automatically from the file provided.'

THE BASICS

Why Fuse?



Eliminate the complexity and inefficiency of managing multiple integrations



Ensure that your integrations are secure, scalable, and easy to maintain



Reduce implementation time and cost



Improve operational efficiency



Enhance data accuracy

How Fuse Works

Integrations in Shopify, made easy.

1. Connect

Choose the systems you want to integrate with.

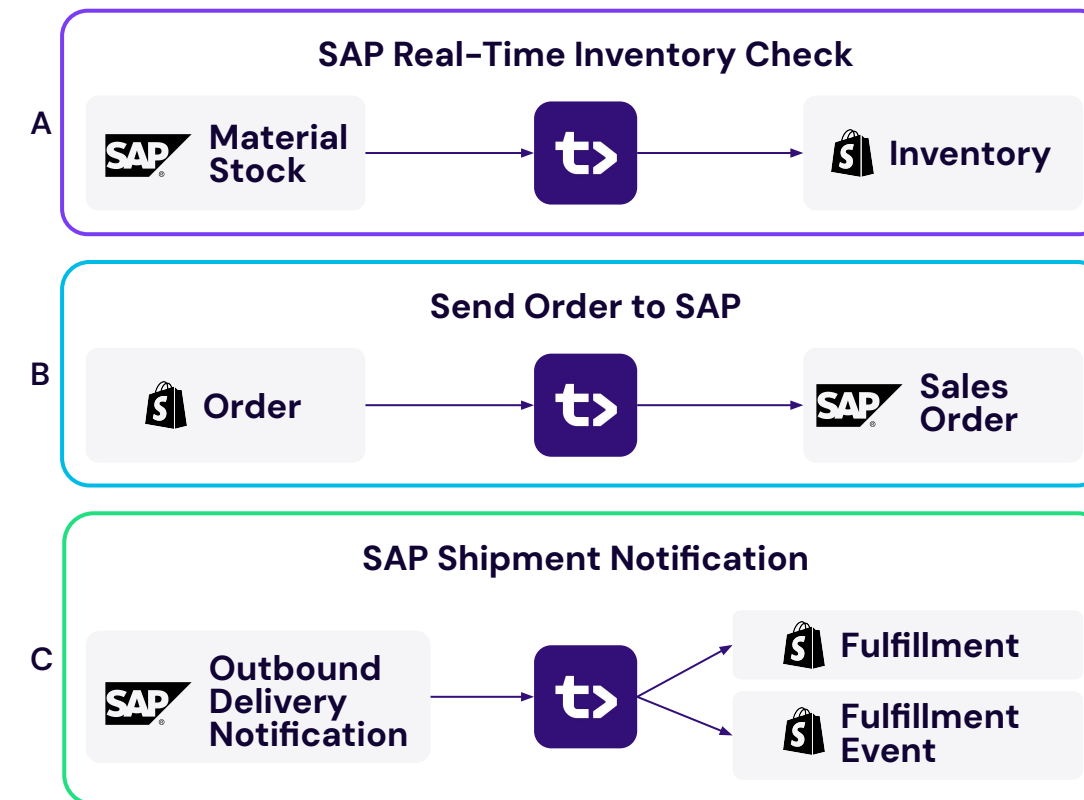
 File Feed Connector

 ERP Connector

Seamlessly connect to industry-standard ERPs like SAP and effortlessly import data from other systems into Shopify, right out of the box.

2. Configure

Pick what you want to integrate.



Effortlessly convert data between systems using the intuitive editor.

3. Activate

Choose where integrations appear across Shopify.



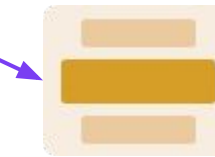
Online Store Themes & Headless



Checkout



Customer Accounts



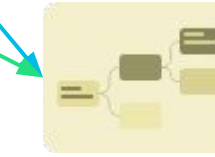
Backend functions



POS



Admin



Flow

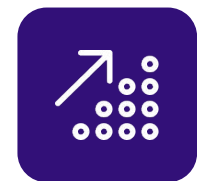
Our Journey to DBOS

Focus on value creation, not managing infrastructure.



Embracing the Serverless Model

We made a decision early on to march towards a serverless architecture to reap the following theoretical benefits:



Scalability: Horizontal scaling to handle peak traffic volume



Cost efficiency: Scaling to zero allows us to save cost during low-traffic times.



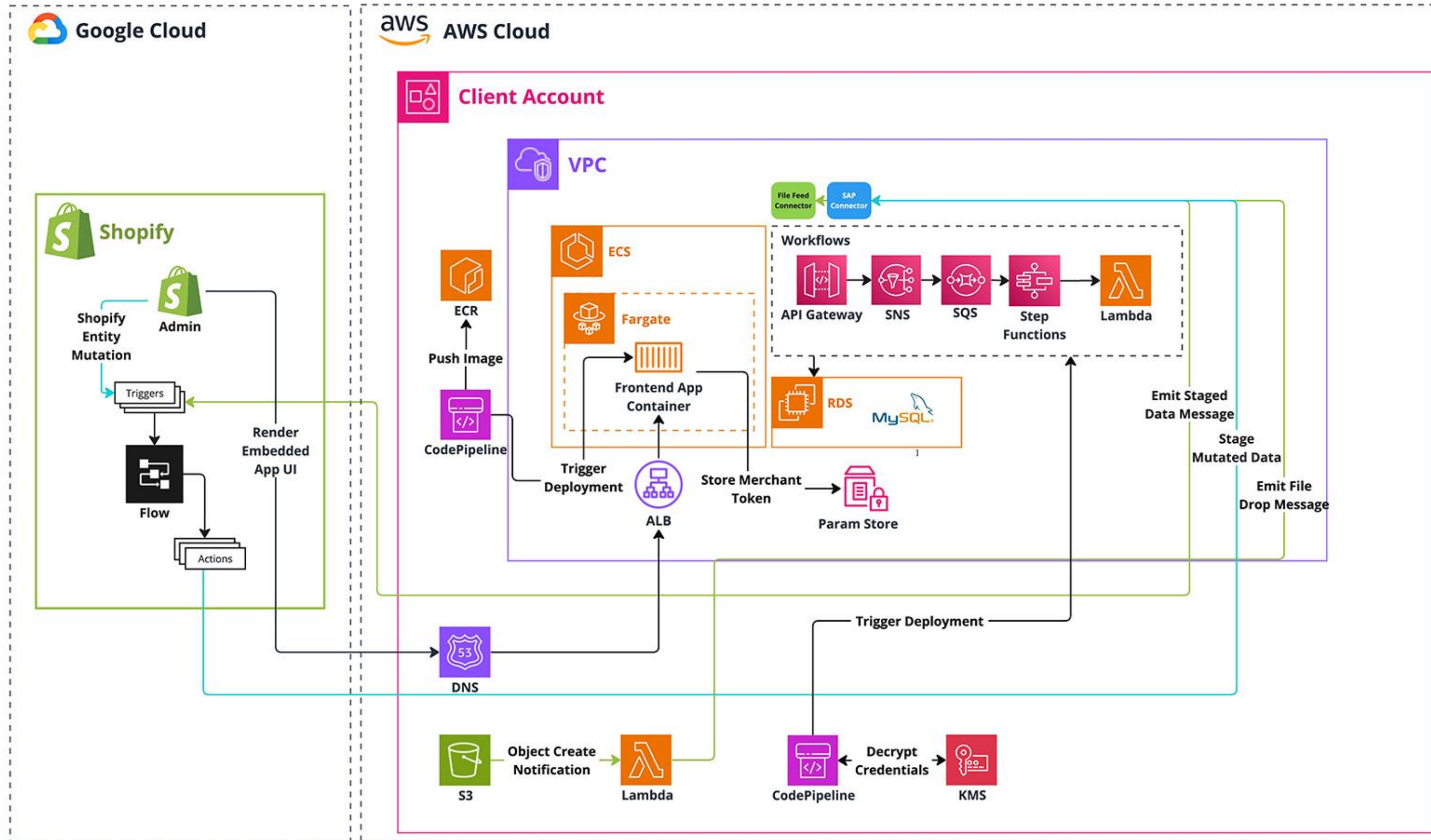
Faster time to market: Focus on core features, not infrastructure



Resilience: Distributed functions across multiple zones, improving reliability

OUR JOURNEY TO DBOS

Vanilla AWS



Why We Chose DBOS



Simplification of Serverless Development

- Eliminated need for IaC, or manual configuration of environment services.
- Development focuses on delivering business value.



Durable Workflows for Business Processes

- Workflows over AWS Step Functions to more easily manage sequencing of operations.

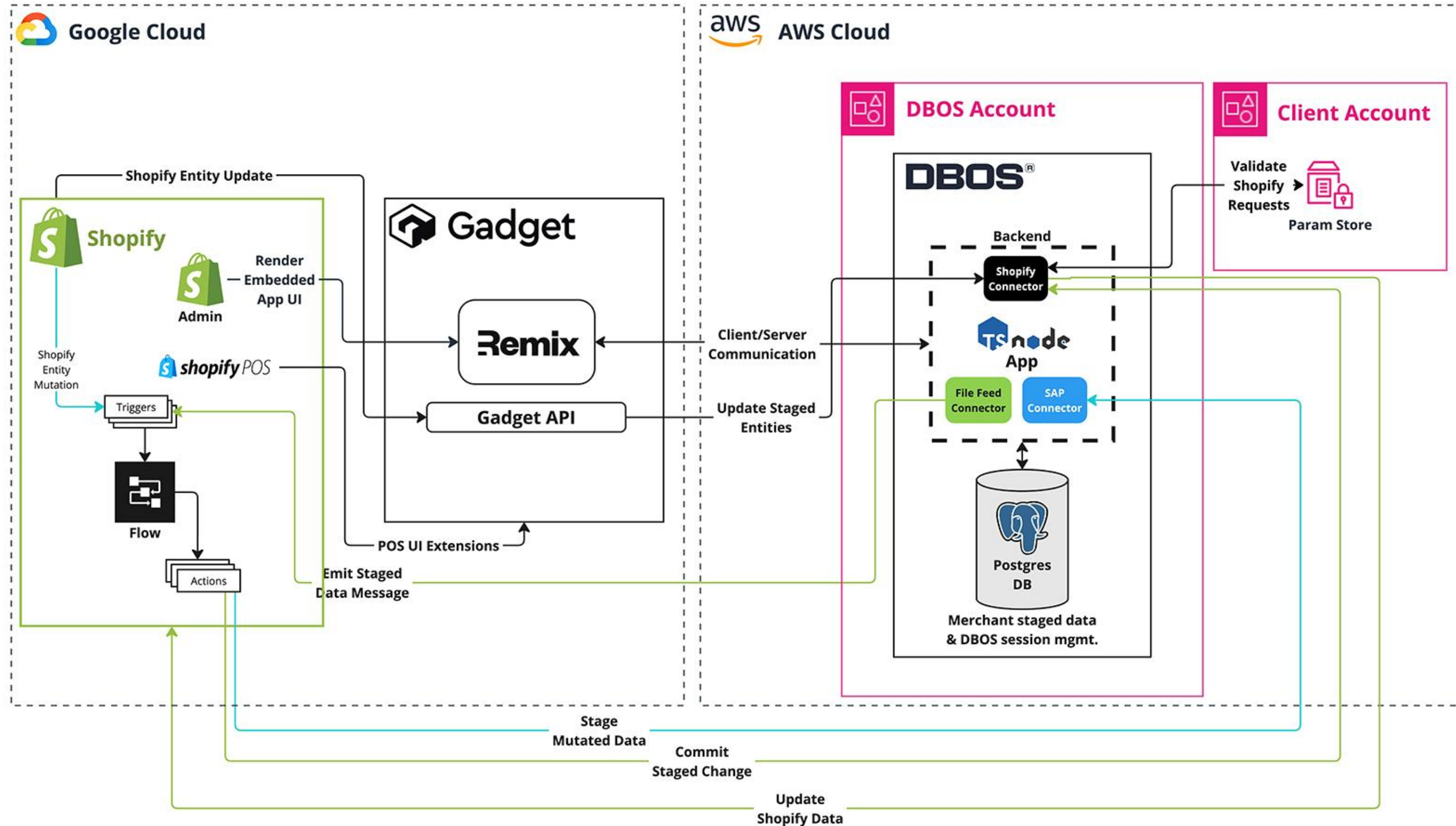


Exactly-Once Processing for Data Integrity

- Transactional idempotency will reduce risk of data duplication
- Easier auditing for tracing transactions end-to-end

OUR JOURNEY TO DBOS

Powered by DBOS



Moving from AWS Lambda to DBOS

Our process to migrate was **FAST**. We were able to get up running on the new DBOS infrastructure in just days. Here's what helped us:

- Spent time to understand DBOS: **Communicators, Transactions, Workflows, etc.**
- Easy to use for replicating our API Gateway routes with simple **decorators**
- Eliminated complex connection/IAM management with DBOS decorators
- Transitioned from Step Functions to Workflows smoothly, similar in paradigm
- Helped us address common pitfalls by monitoring and debugging transactions
 - *Time Traveler is super useful!*
- Allowed us to unify our stack around TypeScript, improving shareability of code/examples across disciplines

Benefits Realized from Switching

Some key highlights we've seen now that we have been operating in our new infrastructure for ~ 5 months:

- **Reduced our Hosting Costs:** By ~15x less than previous AWS costs!
- **Faster Feature Development:** Ability to build and deploy in days, not weeks
- **Improved Performance:** Overall execution times have decreased.
- **Ease of development:** Closer alignment between local and production environments due to the simplistic infrastructure.
- **Quicker Debugging:** Easier troubleshooting through Grafana, Time Traveler and more.
- **Simplified Deployments:** Terraform deployments only required for ancillary services. All codebase deployments now executed solely within Bitbucket Pipelines

Q&A???

DBOS

- Free DBOS Cloud – <https://dbos.dev>
- Source code: <https://github.com/dbos-inc/dbos-transact-py>
- Documentation: <https://docs.dbos.dev/>
- Examples: <https://docs.dbos.dev/examples>

tmg>

- More info: <http://tmg.io>
- TMG Fuse: <https://www.tmg.io/services/fuse/>