Seemless Transitions

Lessons from a Data Migration for a Mission-Critical App

by Ryan Clements, Owner of Byte Bot 🤖

"Database migrations are like moving a house of cards

"Database migrations are like moving a house of cards - do it carefully

"Database migrations are like moving a house of cards - do it carefully, or you'll be picking up pieces for days."

"Database migrations are like moving a house of cards - do it carefully, or you'll be picking up pieces for days."

- ChatGPT

How would you do it?

Goals

Goals

📖 Share a story

Goals

📖 Share a story

🔍 Learn how to identify the need for a data migration

Goals

📖 Share a story

🔍 Learn how to identify the need for a data migration

💭 Learn how to plan a data migration

Goals

📖 Share a story

🔍 Learn how to identify the need for a data migration

💭 Learn how to plan a data migration

🏎️ Learn how to execute a data migration

Goals

📖 Share a story

🔍 Learn how to identify the need for a data migration

💭 Learn how to plan a data migration

🏎️ Learn how to execute a data migration

💣 Learn how to avoid common failure modes

Who am I?

Ryan Clements, Owner of Byte Bot 🤖

Who am I?

Ryan Clements, Owner of Byte Bot 🤖

a full-stack software agency that helps software teams ship valuable software 🚀

Who am I?

Ryan Clements, Owner of Byte Bot 🤖

a full-stack software agency that helps software teams ship valuable software 🚀

Who am I?

Ryan Clements, Owner of Byte Bot 🤖

a full-stack software agency that helps software teams ship valuable software 🚀

- 🌐 bytebot.io
- 📧 info@bytebot.io
- 🐤 @RyanClementsHax
- 📩 bytebot.beehiiv.com/subscribe

Let’s set the scene

Let’s set the scene

The investment firm in question will be unnamed to respect their security policy

The application

The application

🔬 Internal data science tool providing in depth analysis to 1000+ investments

The application

🔬 Internal data science tool providing in depth analysis to 1000+ investments

💪 Enhances the firm’s investment capabilities and getting better every sprint

The application

🔬 Internal data science tool providing in depth analysis to 1000+ investments

💪 Enhances the firm’s investment capabilities and getting better every sprint

👨‍💼 Supports 200+ employees across various departments

Key features

Key features

📈 Instant access to deal information and metrics

Key features

📈 Instant access to deal information and metrics

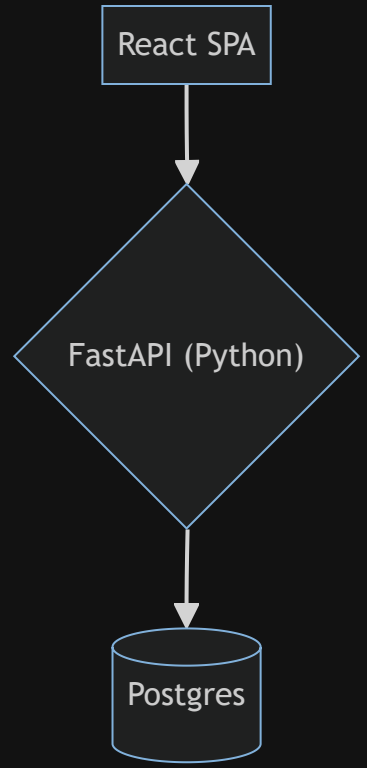
⚙️ Automates routine tasks like monitoring interest rate caps

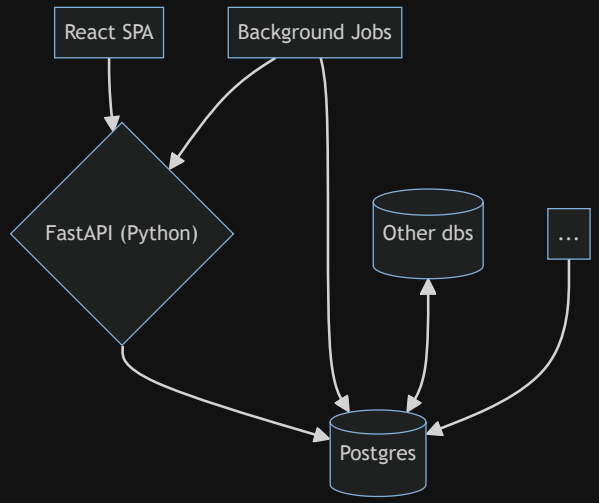
Key features

📈 Instant access to deal information and metrics

⚙️ Automates routine tasks like monitoring interest rate caps

📊 Surfaces fund composition and risk exposures in one place


React SPA
FastAPI (Python)
Postgres


React SPA
Background Jobs
FastAPI (Python)
Other dbs
...
Postgres

Problem: We want to automate document tracking

They all owe us documents regularly

Let’s see how we model the data

Take guarantor data for an example

```
CREATE TABLE loans (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    guarantor TEXT, -- arbitrary text
    -- a ton of other fields ...
);
```

Example data

- Meridian Capital Advisors
- Oakridge Asset Holdings, Zenith Equity Management, Atlas Property Holdings
- Redwood Holdings & Co. & NorthStar Investment Trust
- (a) Fortress Capital Advisors (b) Endeavor Capital Ventures © John Doe
- Atlas Property Holdings; BlueRock Investment Solutions (Pledge)

Example data

- Meridian Capital Advisors
- Oakridge Asset Holdings, Zenith Equity Management, Atlas Property Holdings
- Redwood Holdings & Co. & NorthStar Investment Trust
- (a) Fortress Capital Advisors (b) Endeavor Capital Ventures © John Doe
- Atlas Property Holdings; BlueRock Investment Solutions (Pledge)

Why is this an issue?

```
CREATE TABLE loans (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    guarantor TEXT,
    -- ...
);
```

What we really wanted

```
CREATE TABLE loans (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    -- ...
);
CREATE TABLE guarantors (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    name TEXT,
    loan_id UUID REFERENCES loans(id)
);
```

First, we explored other solutions

```
uuid.uuid5(loan.id, "guarantor-1")
```

Key parts of the alignment document

- What is the context?
- What needs to change?
- What do we gain from this?
- What could we be doing instead?
- What other options are there?
- What are their tradeoffs?
- What committments does this need?
- Do we need to do all of it or can we chunk it?

👍 , but let’s do guarantor data first

Time to plan

Migration strategies

Migration strategies

- Big bang

Migration strategies

- 💥 Big bang
- ✌️ Parallel or dual writes

Migration strategies

- 💥 Big bang
- ✌️ Parallel or dual writes
- 🎂 Phased or incremental

Migration strategies

- 💥 Big bang
- ✌️ Parallel or dual writes
- 🎂 Phased or incremental
- 🔍 Copy and verify

Migration strategies

- 💥 Big bang
- ✌️ Parallel or dual writes
- 🎂 Phased or incremental
- 🔍 Copy and verify
- 🛞 Rolling

Migration strategies

- 💥 Big bang
- ✌️ Parallel or dual writes
- 🎂 Phased or incremental
- 🔍 Copy and verify
- 🛞 Rolling
- 🟦 Blue green deployments

Migration strategies

- 💥 Big bang
- ✌️ Parallel or dual writes
- 🎂 Phased or incremental
- 🔍 Copy and verify
- 🛞 Rolling
- 🟦 Blue green deployments
- 👻 Shadow migration

Migration strategies

- 💥 Big bang
- ✌️ Parallel or dual writes
- 🍰 Phased or incremental
- 🔍 Copy and verify
- 🛞 Rolling
- 🟦 Blue green deployments
- 👻 Shadow migration
- 📨 Event driven

Which one do we use?

Know your application!

Properties of this application

Properties of this application

🔒 Internal app with controlled user access

Properties of this application

- 🔓 Internal app with controlled user access
- 🕒 Users clocked out at the end of the week, allowing for downtime

Properties of this application

- 🔓 Internal app with controlled user access
- 🕒 Users clocked out at the end of the week, allowing for downtime
- 📉 Low-traffic features, reducing complexity during the migration

Properties of this application

- 🔓 Internal app with controlled user access
- 🕒 Users clocked out at the end of the week, allowing for downtime
- 📉 Low-traffic features, reducing complexity during the migration
- ✊ Limited data size, allowing manual verification

Impacted features

Impacted features

💻 Excel onboarding

Impacted features

💻 Excel onboarding

🧮 Needed full CRUD over guarantors

Impacted features

- 💻 Excel onboarding
- 🧮 Needed full CRUD over guarantors
- 🧥 Foreign Data Wrapper (FDW)

Impacted features

- 💻 Excel onboarding
- 🧮 Needed full CRUD over guarantors
- 🧥 Foreign Data Wrapper (FDW)
- 📩 Reporting export

What do you think?

We went with big bang  (but technically phased)

What goes into this document?

What goes into this document?

- Why you’re doing it

What goes into this document?

- Why you’re doing it
- What it buys you

What goes into this document?

- Why you’re doing it
- What it buys you
- What features are impacted

What goes into this document?

- Why you’re doing it
- What it buys you
- What features are impacted
- What is the data (don’t assume people know)

What goes into this document?

- Why you’re doing it
- What it buys you
- What features are impacted
- What is the data (don’t assume people know)
- What needs to be changed and by whom?

What goes into this document?

- Why you’re doing it
- What it buys you
- What features are impacted
- What is the data (don’t assume people know)
- What needs to be changed and by whom?
- What dependencies are there?

What goes into this document?

- Why you’re doing it
- What it buys you
- What features are impacted
- What is the data (don’t assume people know)
- What needs to be changed and by whom?
- What dependencies are there?
- How will the actual migration happen?

What goes into this document?

- Why you're doing it
- What it buys you
- What features are impacted
- What is the data (don't assume people know)
- What needs to be changed and by whom?
- What dependencies are there?
- How will the actual migration happen?
- When will we do this?

What goes into this document?

- Why you're doing it
- What it buys you
- What features are impacted
- What is the data (don't assume people know)
- What needs to be changed and by whom?
- What dependencies are there?
- How will the actual migration happen?
- When will we do this?
- What steps does it need to happen in?

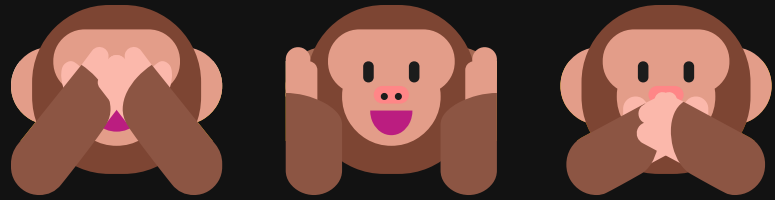
What goes into this document?

- Why you're doing it
- What it buys you
- What features are impacted
- What is the data (don't assume people know)
- What needs to be changed and by whom?
- What dependencies are there?
- How will the actual migration happen?
- When will we do this?
- What steps does it need to happen in?
- Are there any dependencies to deployment?

What goes into this document?

- Why you're doing it
- What it buys you
- What features are impacted
- What is the data (don't assume people know)
- What needs to be changed and by whom?
- What dependencies are there?
- How will the actual migration happen?
- When will we do this?
- What steps does it need to happen in?
- Are there any dependencies to deployment?
- How will you verify it? (Cannot stress how important this is)

We got everything done in a sprint and there weren’t any issues



The dragons

The dragons

❌ Making schema changes wasn’t well documented eating a lot of time

The dragons

- ❌ Making schema changes wasn’t well documented eating a lot of time
- ❌ Few people knew how the FDW worked or how to develop/test changes to it

The dragons

- ❌ Making schema changes wasn’t well documented eating a lot of time
- ❌ Few people knew how the FDW worked or how to develop/test changes to it
- ❌ And one other one…

Key take aways

Key take aways

- 😫 Avoid them if you can

Key take aways

- 😫 Avoid them if you can
- 🤦‍♀️ Understand your application and users

Key take aways

- 😫 Avoid them if you can
- 🤦‍♀️ Understand your application and users
- 💊 Choose the right medicine

Key take aways

- 😫 Avoid them if you can
- 🤦‍♀️ Understand your application and users
- 💊 Choose the right medicine
- 💪 Most of the effort goes into the nontech parts

Key take aways

- 😫 Avoid them if you can
- 👩‍💼 Understand your application and users
- 💊 Choose the right medicine
- 💪 Most of the effort goes into the nontech parts
- 🗣️ You can’t overcommunicate

Key take aways

- 😫 Avoid them if you can
- 👩‍💼 Understand your application and users
- 💊 Choose the right medicine
- 💪 Most of the effort goes into the nontech parts
- 🗣️ You can’t overcommunicate
- 💥 Expect there to be bumps, and the only way to find all of them is to actually do the work

Key take aways

- 😫 Avoid them if you can
- 👩‍💼 Understand your application and users
- 💊 Choose the right medicine
- 💪 Most of the effort goes into the nontech parts
- 🗣️ You can’t overcommunicate
- 💥 Expect there to be bumps, and the only way to find all of them is to actually do the work
- 🙈 Even the directly responsible engineers don’t always fully understand the system

Got questions?

SCAN ME

Here’s how we can stay in touch

- bytebot.io
- info@bytebot.io
- @RyanClementsHax
- bytebot.beehiiv.com/subscribe