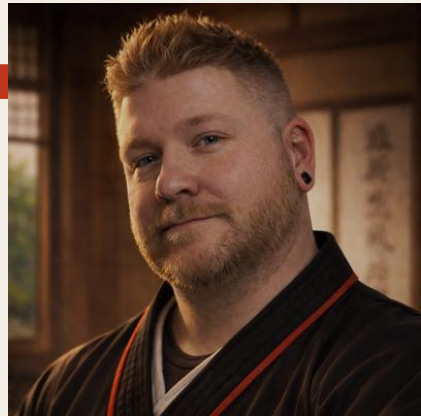
DESIGNING DATA PIPELINES

THAT DON'T HATE YOU

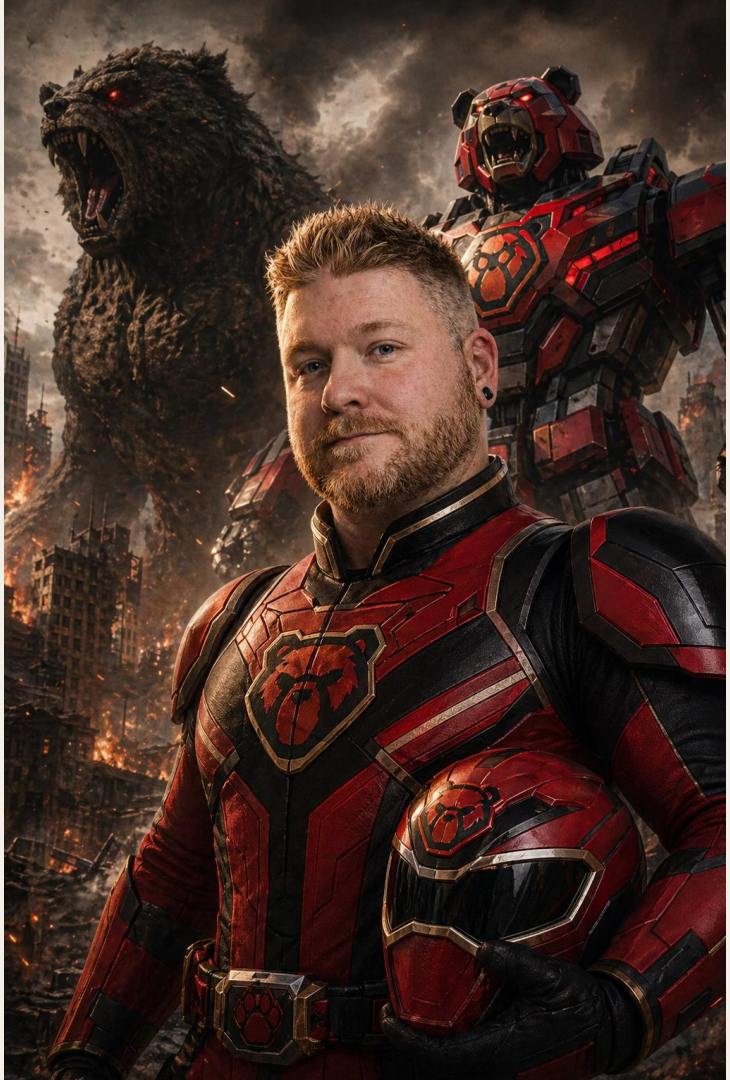
SIX MONTHS LATER

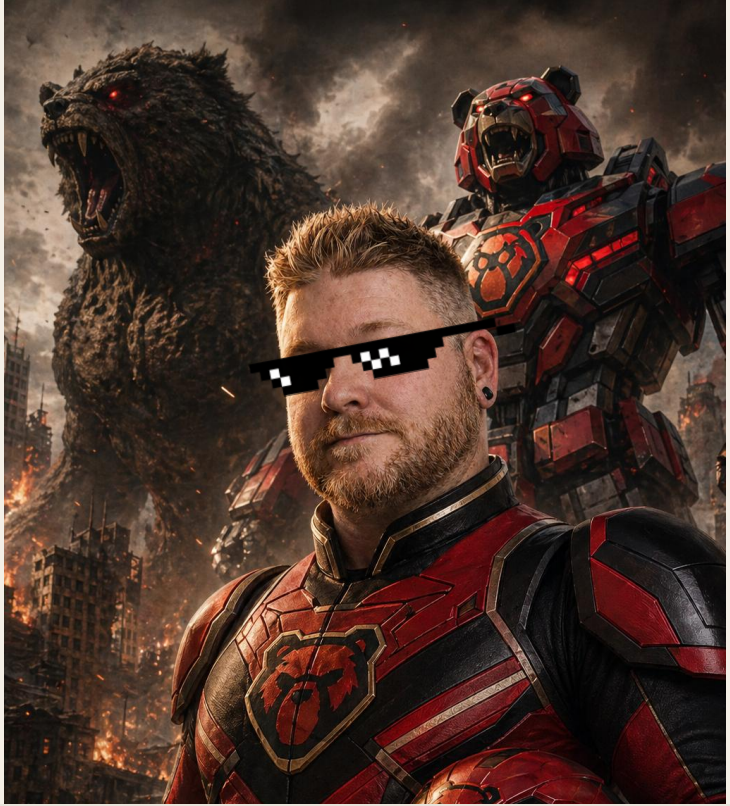
A Sci-Fi Kung Fu Inspired Story About Data Engineering Fundamentals



Chris Birie | Stir Trek

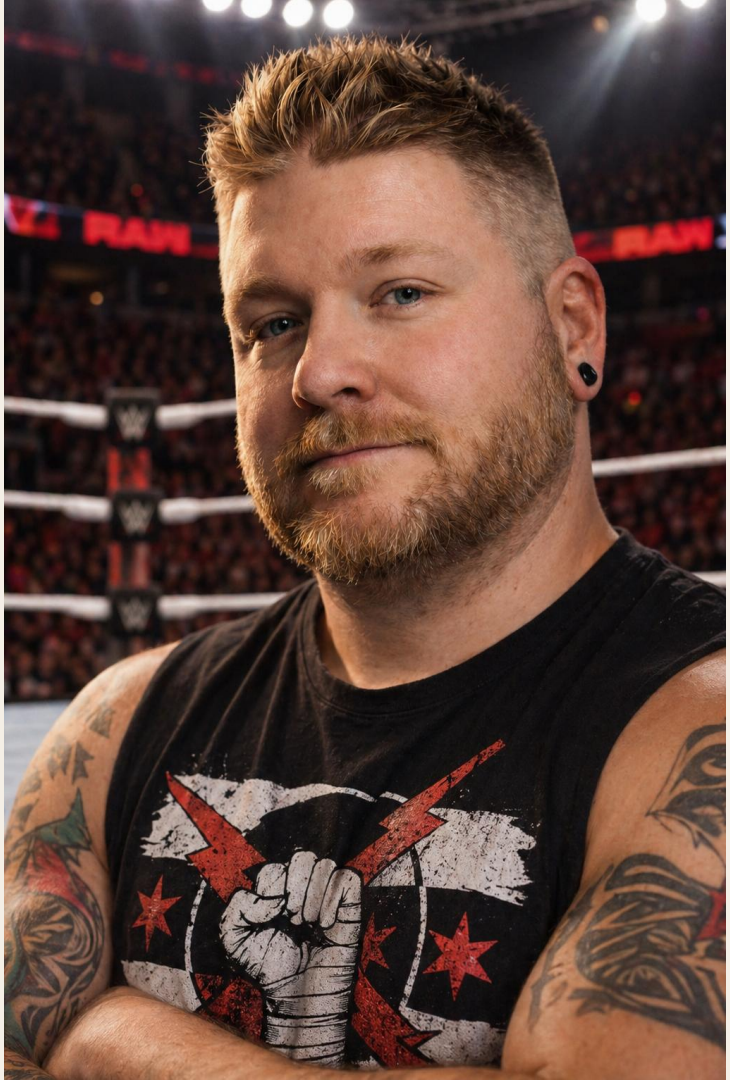
ABOUT ME

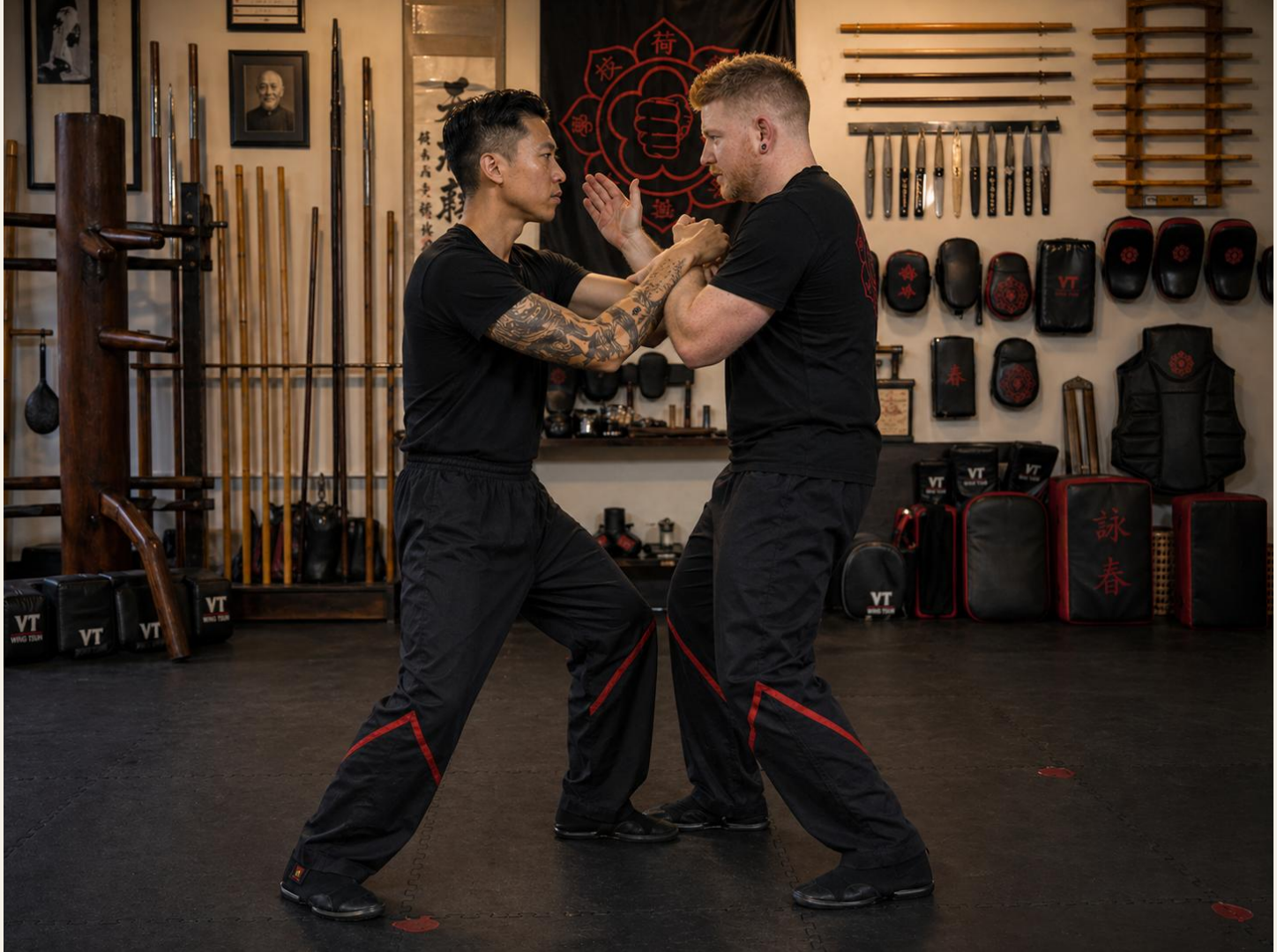


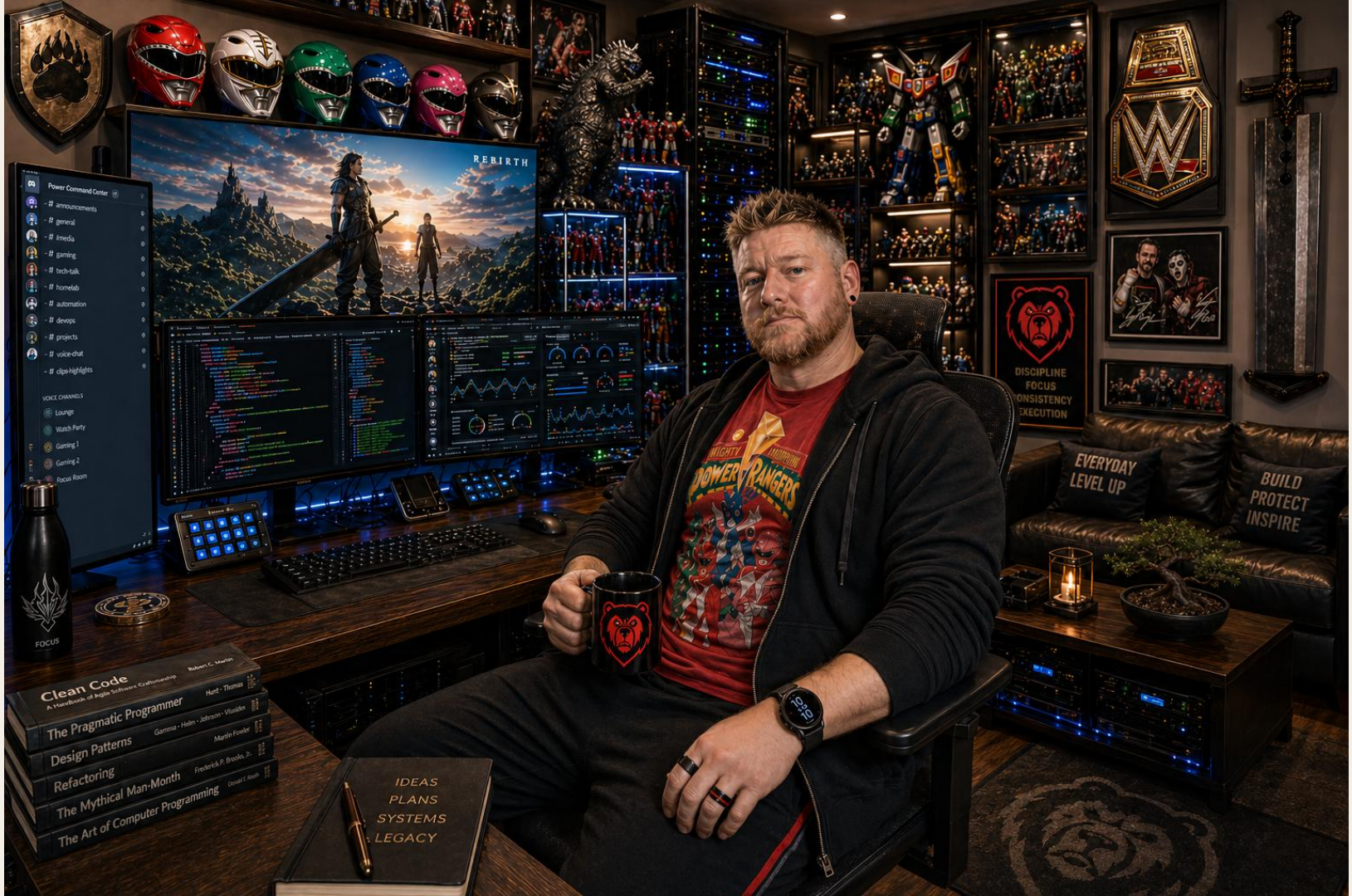


YES I USED AI TO GENERATE THESE ALL PHOTOS IN THIS PRESENTATION…

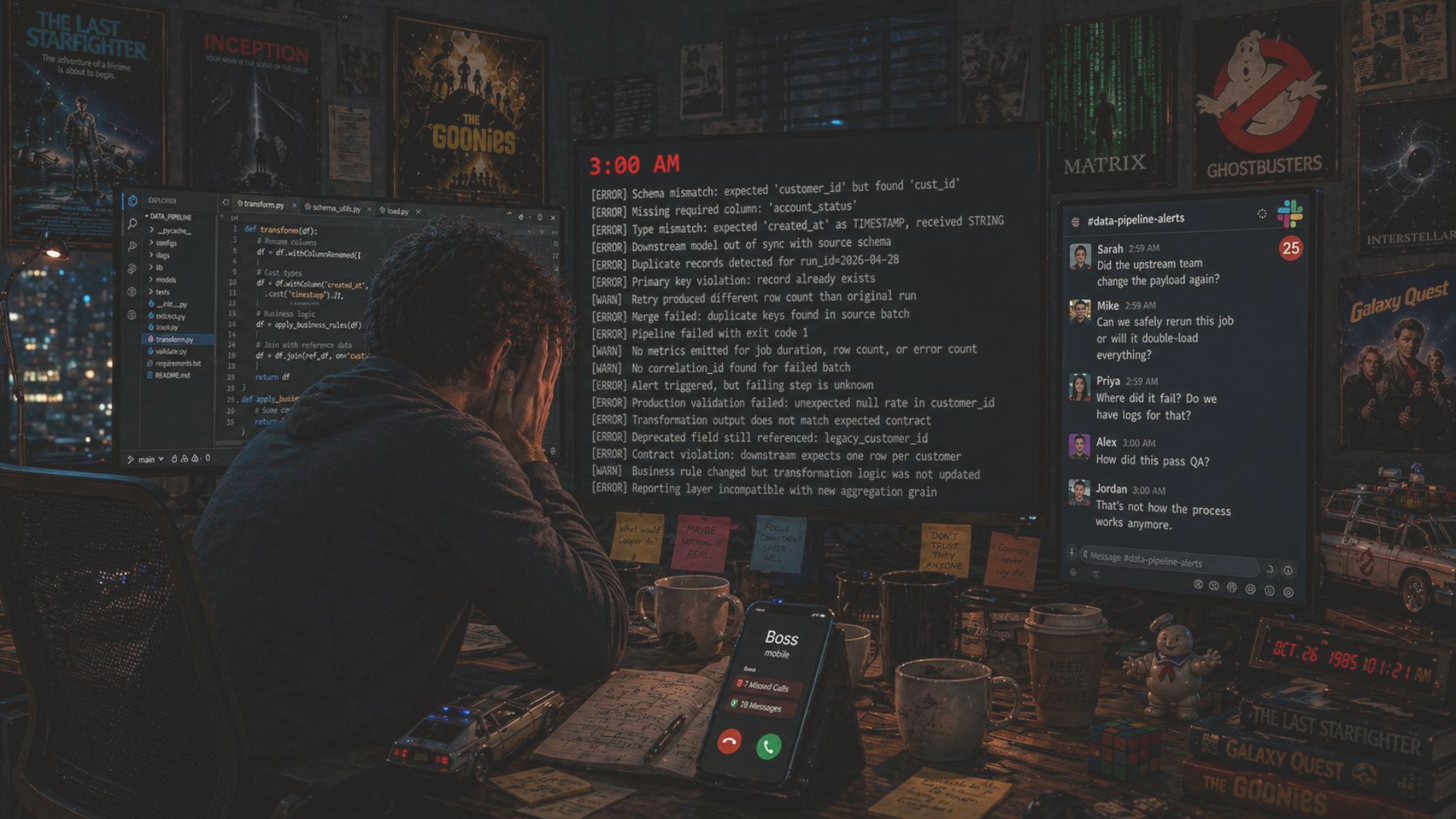
DEAL WITH IT!!!

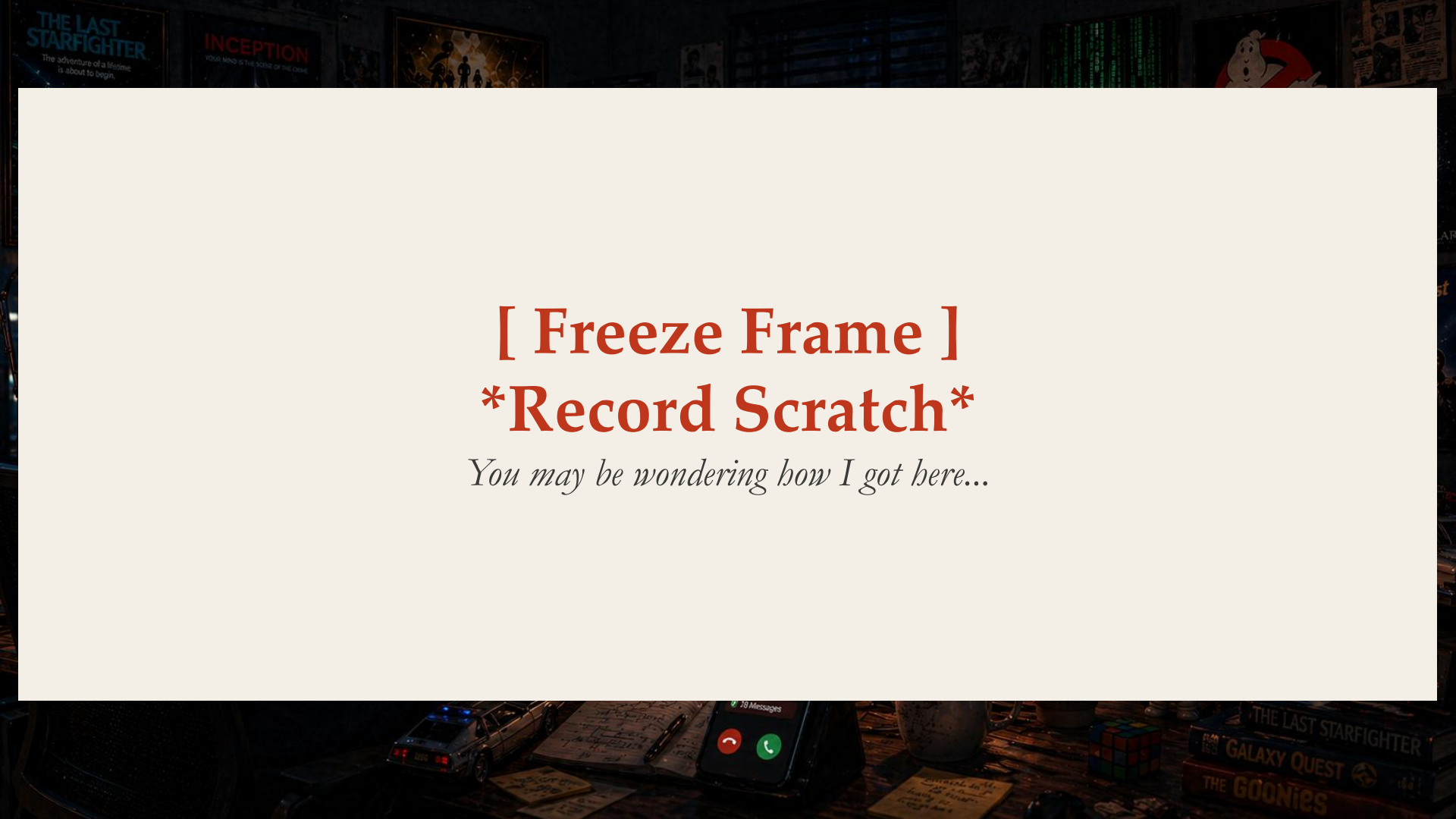

RAW
RAW


VT
VT
VT
VT
VT
VT
VT
VT
詠
春


REBIRTH
Power Command Center
announcements
general
media
gaming
tech-talk
homelab
automation
devops
projects
voice-chat
clips-highlights
VOICE CHANNELS
Lounge
Watch Party
Gaming 1
Gaming 2
Focus Room
FOCUS
DISCIPLINE
FOCUS
CONSISTENCY
EXECUTION
EVERYDAY
LEVEL UP
BUILD
PROTECT
INSPIRE
MIGHTY
MORPHIN
POWER RANGERS
Clean Code
A Handbook of Agile Software Craftsmanship
Robert C. Martin
The Pragmatic Programmer
Hunt · Thomas
Design Patterns
Gamma · Helm · Johnson · Vlissides
Refactoring
Martin Fowler
The Mythical Man-Month
Frederick P. Brooks, Jr.
The Art of Computer Programming
Donald E. Knuth
IDEAS
PLANS
SYSTEMS
LEGACY

Data Pipelines! Specifically great data pipelines that don't hate you six months later!


3:00 AM
[ERROR] Schema mismatch: expected 'customer_id' but found 'cust_id'
[ERROR] Missing required column: 'account_status'
[ERROR] Type mismatch: expected 'created_at' as TIMESTAMP, received STRING
[ERROR] Downstream model out of sync with source schema
[ERROR] Duplicate records detected for run_id=2026-04-28
[ERROR] Primary key violation: record already exists
[WARN] Retry produced different row count than original run
[ERROR] Merge failed: duplicate keys found in source batch
[ERROR] Pipeline failed with exit code 1
[WARN] No metrics emitted for job duration, row count, or error count
[WARN] No correlation_id found for failed batch
[ERROR] Alert triggered, but failing step is unknown
[ERROR] Production validation failed: unexpected null rate in customer_id
[ERROR] Transformation output does not match expected contract
[ERROR] Deprecated field still referenced: legacy_customer_id
[ERROR] Contract violation: downstream expects one row per customer
[WARN] Business rule changed but transformation logic was not updated
[ERROR] Reporting layer incompatible with new aggregation grain
#data-pipeline-alerts
25
Sarah 2:59 AM
Did the upstream team change the payload again?
Mike 2:59 AM
Can we safely rerun this job or will it double-load everything?
Priya 2:59 AM
Where did it fail? Do we have logs for that?
Alex 3:00 AM
How did this pass QA?
Jordan 3:00 AM
That's not how the process works anymore.
Message #data-pipeline-alerts
Boss
mobile
7 Missed Calls
78 Messages
THE LAST STARFIGHTER
INCEPTION
THE GOONIES
MATRIX
GHOSTBUSTERS
INTERSTELLAR
Galaxy Quest
What would Cooper do?
MAYBE NOTHING IS REAL
FOCUS COMMITMENT SHEER WILL
DON'T TRUST THEY ANYONE
Goonies never say die.
NEED MORE COFFEE
OCT 26 1985 10:21 AM

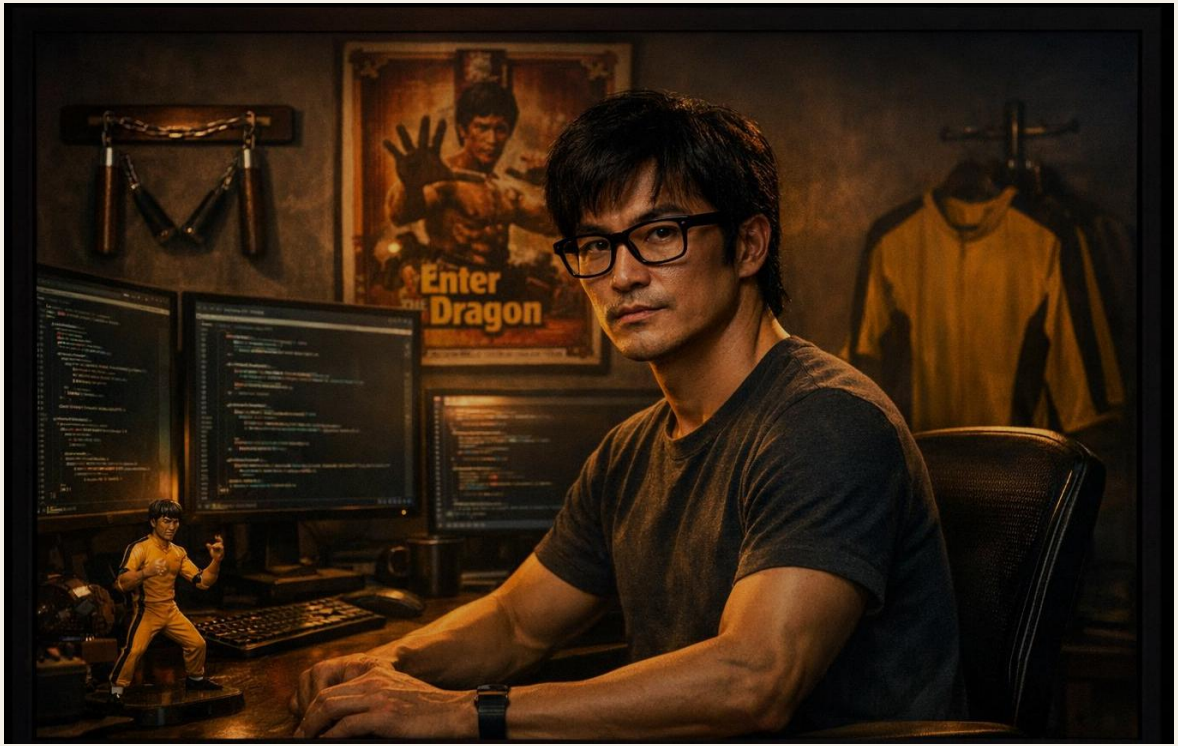


[Freeze Frame]
Record Scratch

You may be wondering how I got here...

SIX MONTHS EARLIER...

Enter the ~~Dragon~~ Bruce.



Bruce was brilliant.

Bruce was fast.

Bruce used AI.

Bruce shipped early.

The pipeline worked beautifully

Everyone was happy.

Bruce got a pat on the back.

And probably a promotion.

Bruce moved on. Bruce is not here anymore.

This was Bruce's pipeline.

It is now your problem.

Pipelines built on shaky fundamentals don't fail on day one.
They fail when pressure arrives.

BRUCE DIDN'T BUILD A BAD PIPELINE.

He built a pipeline that worked perfectly for the fight he trained for.

The problem is the fight changed.

"The pipeline didn't fail because Bruce was lazy or bad at his job.
It failed because nobody taught him the fundamentals, or they were ignored.
And then Bruce was gone."

Bruce is gone. But his pipeline lives on.

And right now, at 3am, it is absolutely on fire.

BUT WHAT IF
WE COULD GO BACK?

Six months ago, Bruce was building the pipeline.
He didn't know what was coming. Nobody did.
But what if we could travel back and change the outcome?
What would we teach him?

Every lesson ahead is one Bruce never learned.
Each principle — a mistake he made so you don't have to.

Five lessons. Five chances to change the outcome.

THE FIVE LESSONS BRUCE NEVER LEARNED

01

SCHEMA EVOLUTION

Don't Build Your Stance on Sand

02

IDEMPOTENCY

You think a fight is one throw, one kick?

03

OBSERVABILITY

Don't charge him blindly. You've got to listen. Listen.

04

TESTABILITY

What was that? An exhibition?

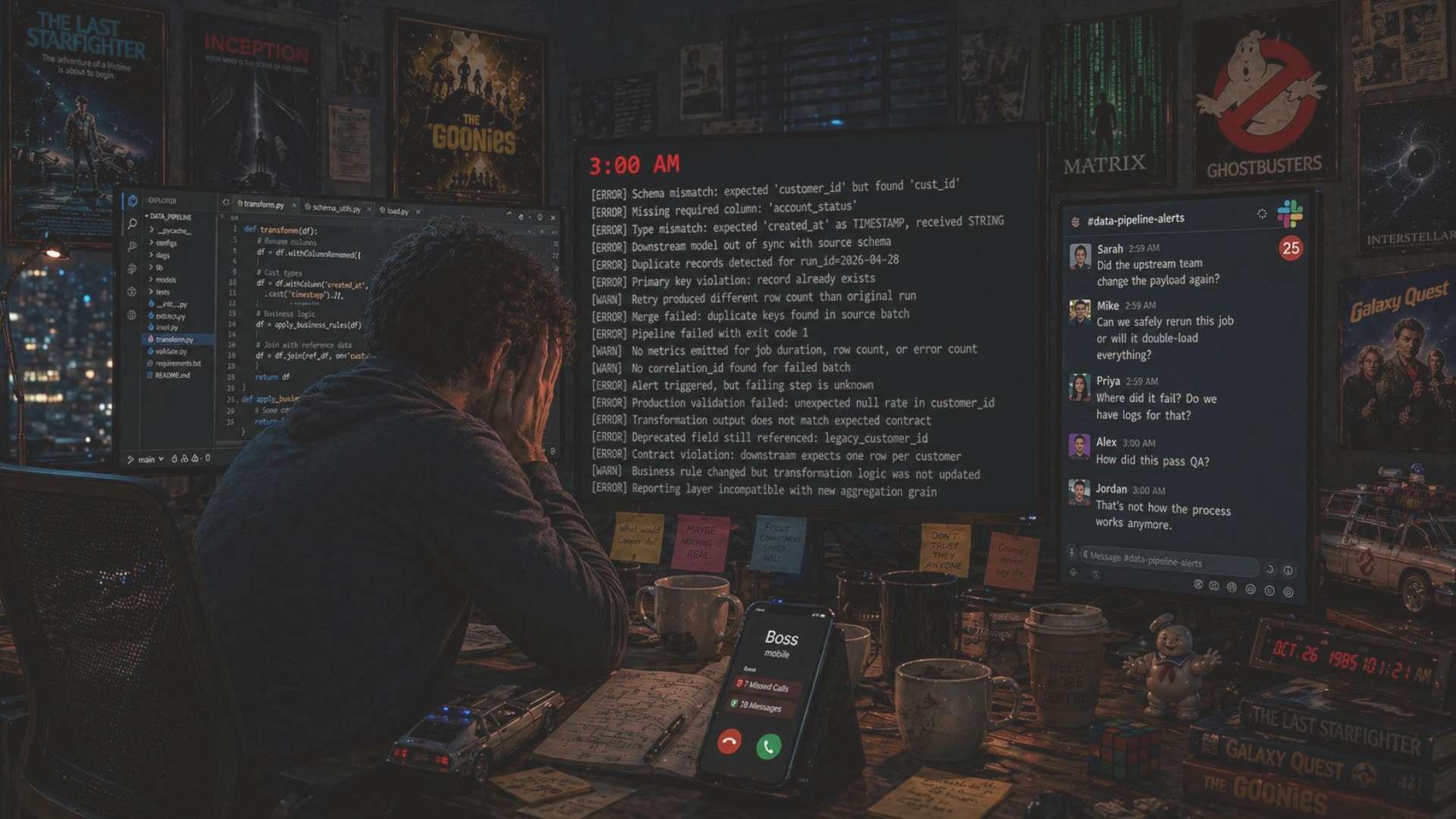
05

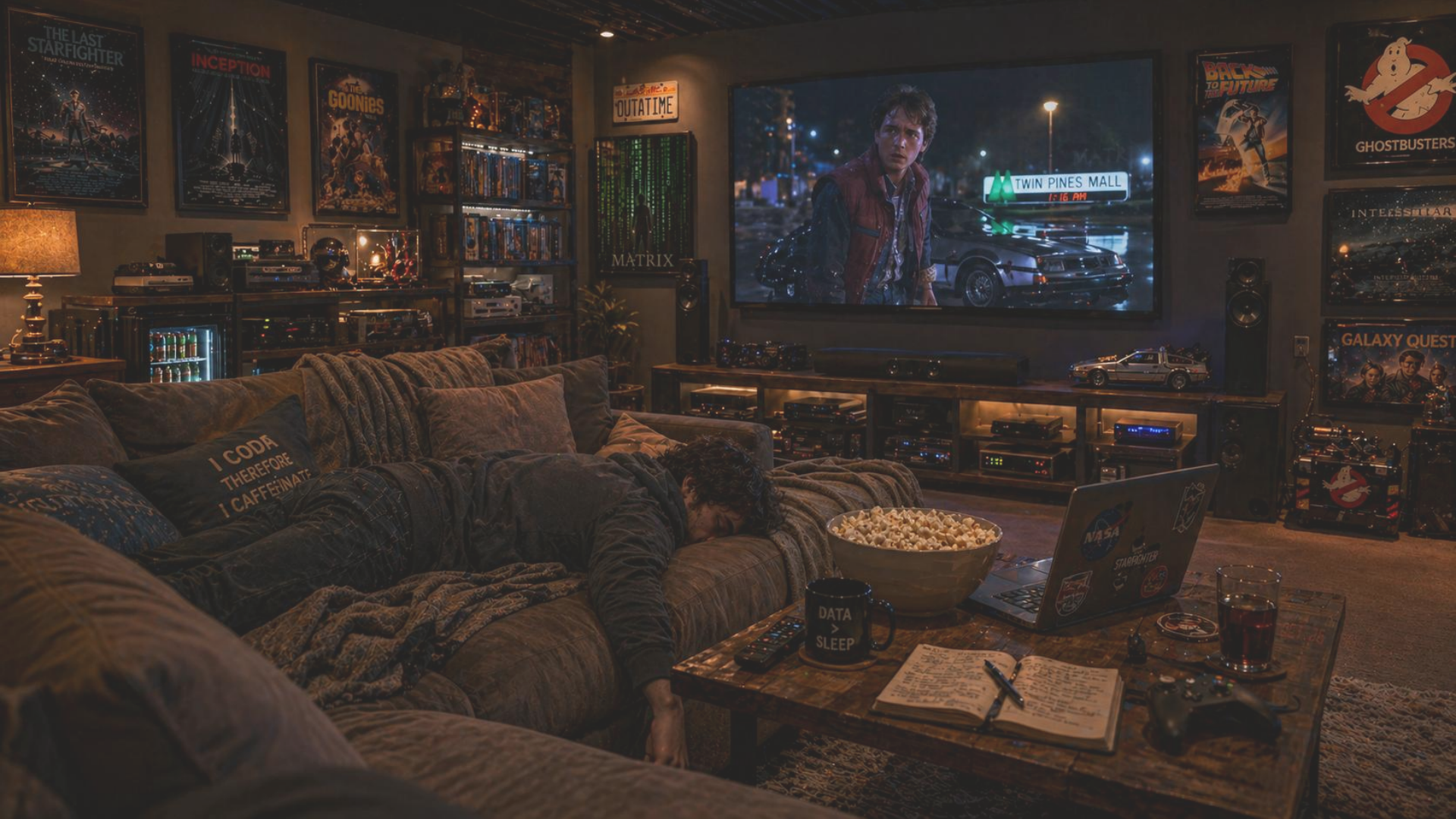
DESIGN FOR CHANGE

Be Water My Friend

"A warrior who only trains for the tournament
will fall when the real fight comes."

— Some Wise Sifu (Probably)


3:00 AM
[ERROR] Schema mismatch: expected 'customer_id' but found 'cust_id'
[ERROR] Missing required column: 'account_status'
[ERROR] Type mismatch: expected 'created_at' as TIMESTAMP, received STRING
[ERROR] Downstream model out of sync with source schema
[ERROR] Duplicate records detected for run_id=2026-04-28
[ERROR] Primary key violation: record already exists
[WARN] Retry produced different row count than original run
[ERROR] Merge failed: duplicate keys found in source batch
[ERROR] Pipeline failed with exit code 1
[WARN] No metrics emitted for job duration, row count, or error count
[WARN] No correlation_id found for failed batch
[ERROR] Alert triggered, but failing step is unknown
[ERROR] Production validation failed: unexpected null rate in customer_id
[ERROR] Transformation output does not match expected contract
[ERROR] Deprecated field still referenced: legacy_customer_id
[ERROR] Contract violation: downstream expects one row per customer
[WARN] Business rule changed but transformation logic was not updated
[ERROR] Reporting layer incompatible with new aggregation grain
#data-pipeline-alerts
25
Sarah 2:59 AM
Did the upstream team change the payload again?
Mike 2:59 AM
Can we safely rerun this job or will it double-load everything?
Priya 2:59 AM
Where did it fail? Do we have logs for that?
Alex 3:00 AM
How did this pass QA?
Jordan 3:00 AM
That's not how the process works anymore.
Message #data-pipeline-alerts
Boss
mobile
7 Missed Calls
78 Messages
THE LAST STARFIGHTER
INCEPTION
THE GOONIES
MATRIX
GHOSTBUSTERS
INTERSTELLAR
Galaxy Quest
What would Cooper do?
MAYBE NOTHING IS REAL
FOCUS COMMITMENT SHEER WILL
DON'T TRUST THEY ANYONE
Goonies never say die.
NEED MORE COFFEE
OCT 26 1985 10:21 AM


THE LAST STARFIGHTER
INCEPTION
THE GOONIES
OUTATIME
MATRIX
TWIN PINES MALL
1:16 AM
BACK TO THE FUTURE
GHOSTBUSTERS
GALAXY QUEST
I CODA
THEREFORE
I CAFFEINATE
DATA
>
SLEEP
NASA
STARFIGHTER

WHAT IF…?

LESSON 01

SCHEMA EVOLUTION

"Don't Build Your Stance on Sand"

BRUCE'S MISTAKE

6 months ago: Bruce trusted the upstream system to not change, and they didn't change during warranty.

Today: The source system changed their database/file layout and source types but didn't let us know. They also sent us some corrupted columns that should have had dates, but it had gibberish

Result: Bruce's pipeline exploded not so silently with an array of errors

THE LESSON

"Adapt what is useful. Reject what is useless."

Bruce trusted what the source sent him and never built a system to question it.

Adapt what is useful — accept valid data, handle schema changes gracefully, version your contracts.

Reject what is useless — validate at the boundary, route bad records before they corrupt what's downstream.

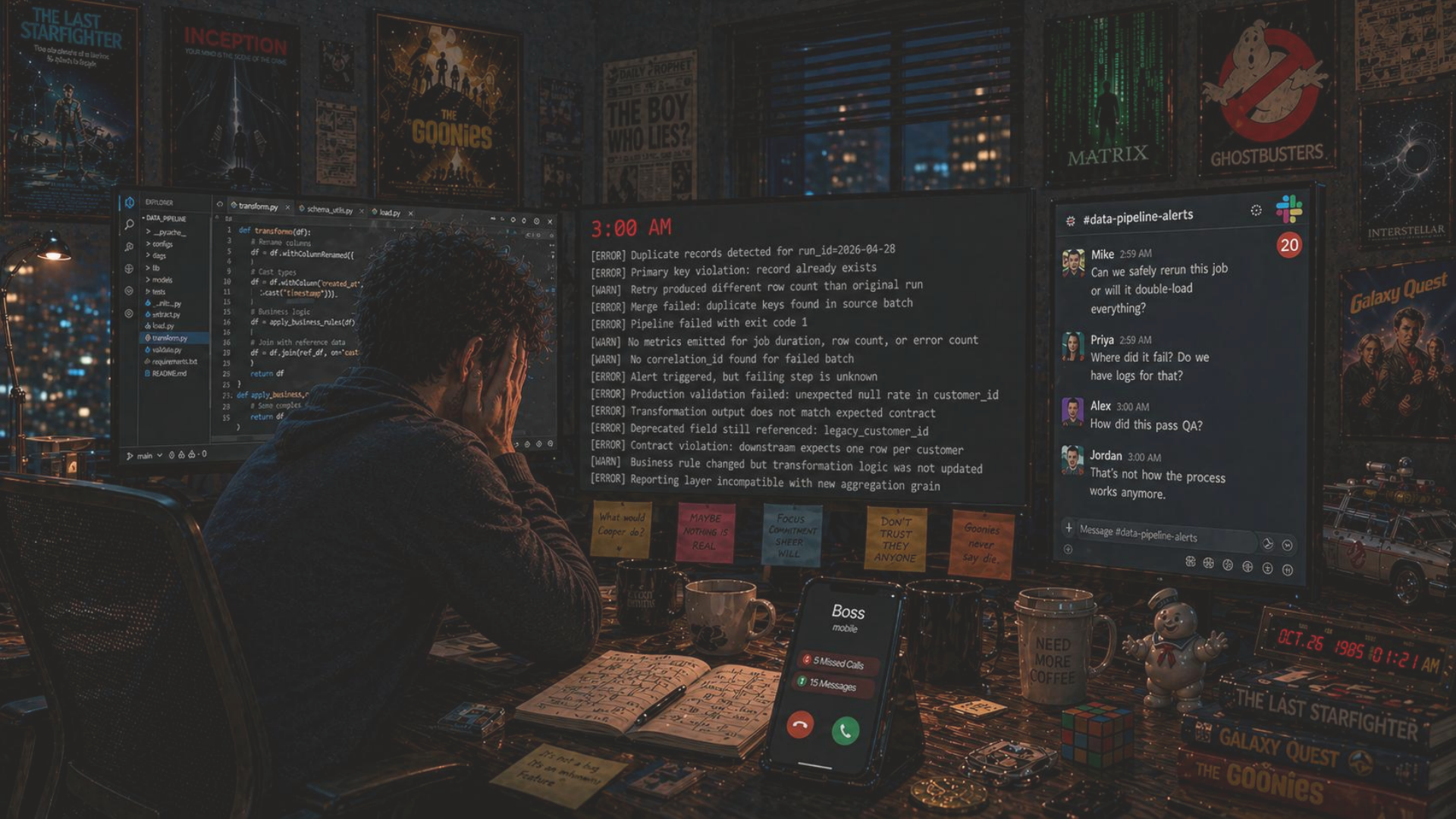
Your pipeline's job is to be paranoid at the front door so nothing weird makes it into the house.

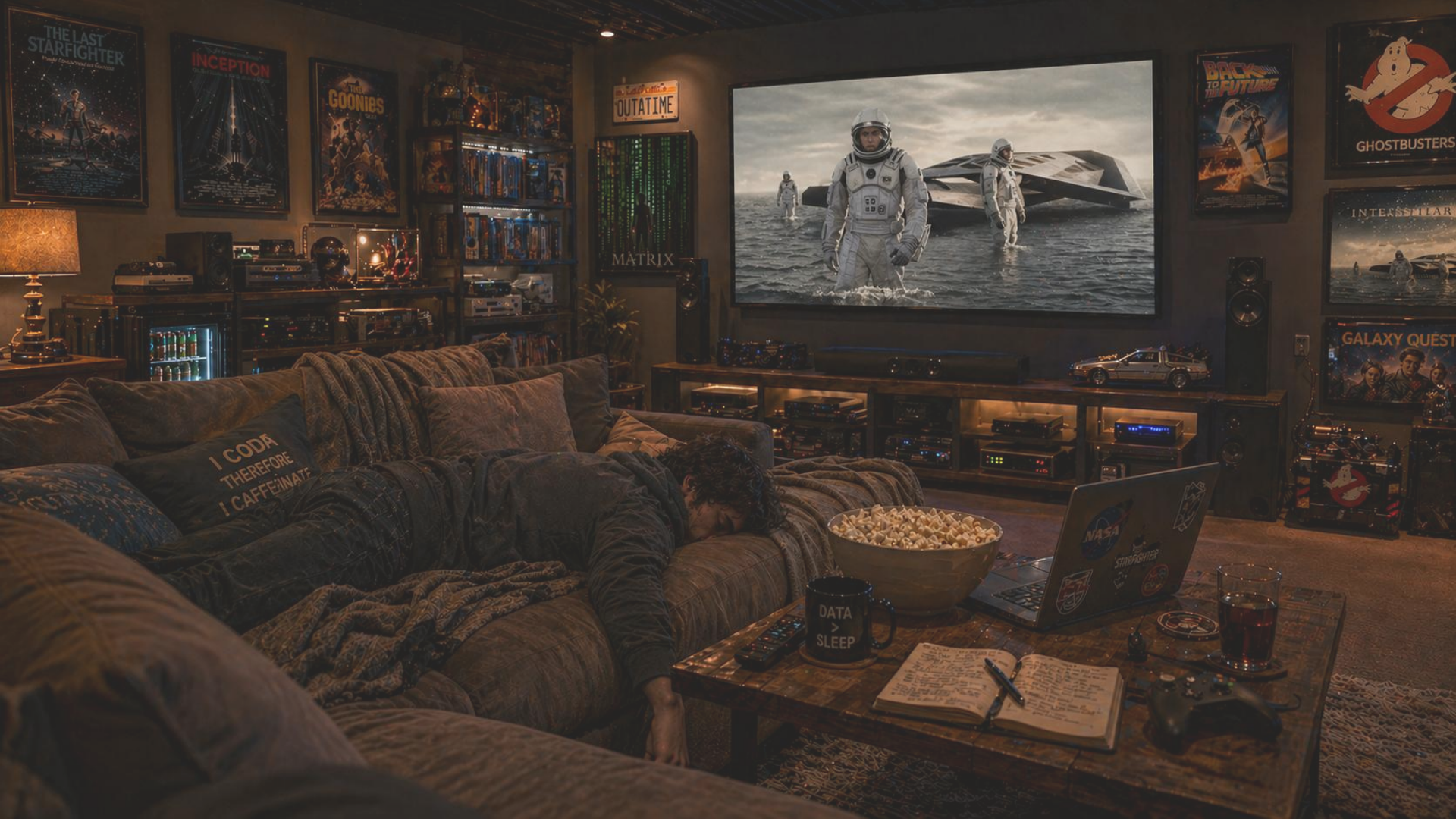
THE FIX

- Nullable fields by default/default values in schema definition
- Versioned schema contracts that are checked regularly
- Defensive parsing/data validation
- Never blindly trust the source, trust but verify

"Your pipeline will outlive the person who designed the source system. Plan for that."

THE PAST HAS CHANGED!


THE LAST STARFIGHTER
INCEPTION
THE GOONIES
THE BOY WHO LIES?
MATRIX
GHOSTBUSTERS
INTERSTELLAR
Galaxy Quest
EXPLORER
DATA_PIPELINE
transform.py
schema_utils.py
load.py
def transform(df):
# Rename columns
df = df.withColumnRenamed({
}
# Cast types
df = df.withColumn('created_at',
.cast("timestamp"))
}
# Business logic
df = apply_business_rules(df)
}
# Join with reference data
df = df.join(ref_df, on="cust
}
return df
}
def apply_business_r
Some complex
return df
}
main
3:00 AM
[ERROR] Duplicate records detected for run_id=2026-04-28
[ERROR] Primary key violation: record already exists
[WARN] Retry produced different row count than original run
[ERROR] Merge failed: duplicate keys found in source batch
[ERROR] Pipeline failed with exit code 1
[WARN] No metrics emitted for job duration, row count, or error count
[WARN] No correlation_id found for failed batch
[ERROR] Alert triggered, but failing step is unknown
[ERROR] Production validation failed: unexpected null rate in customer_id
[ERROR] Transformation output does not match expected contract
[ERROR] Deprecated field still referenced: legacy_customer_id
[ERROR] Contract violation: downstream expects one row per customer
[WARN] Business rule changed but transformation logic was not updated
[ERROR] Reporting layer incompatible with new aggregation grain
#data-pipeline-alerts
20
Mike 2:59 AM
Can we safely rerun this job or will it double-load everything?
Priya 2:59 AM
Where did it fail? Do we have logs for that?
Alex 3:00 AM
How did this pass QA?
Jordan 3:00 AM
That's not how the process works anymore.
Message #data-pipeline-alerts
What would Cooper do?
MAYBE NOTHING IS REAL
FOCUS COMMITMENT SHEER WILL
DON'T TRUST THEY ANYONE
Goonies never say die.
Boss
mobile
5 Missed Calls
15 Messages
NEED MORE COFFEE
OCT 26 1985 01:21 AM
THE LAST STARFIGHTER
GALAXY QUEST
THE GOONIES


THE LAST STARFIGHTER
INCEPTION
THE GOONIES
OUTATIME
MATRIX
BACK TO THE FUTURE
GHOSTBUSTERS
INTERSTELLAR
GALAXY QUEST
I CODA THEREFORE I CAFFEINATE
DATA > SLEEP
NASA
STARFIGHTER

WHAT IF…?

LESSON 02

IDEMPOTENCY

"You think a fight is one throw, one kick?"

BRUCE'S MISTAKE

6 months ago: Bruce built the pipeline to run once per day, start to finish, no interruptions. That was the plan. That assumption was baked into every line of code.

Today: The target database went down halfway through the load. The support team brought it back up and re-ran the pipeline to finish the job.

Result: Primary key violations. Duplicate records. Audit flags on doubled counts. The fix took longer than the original outage.

THE LESSON

"Expect to be hit more than once. The fighter who trains for one clean exchange is not prepared for the real fight."

A pipeline that can only safely run once is a pipeline waiting to fail.

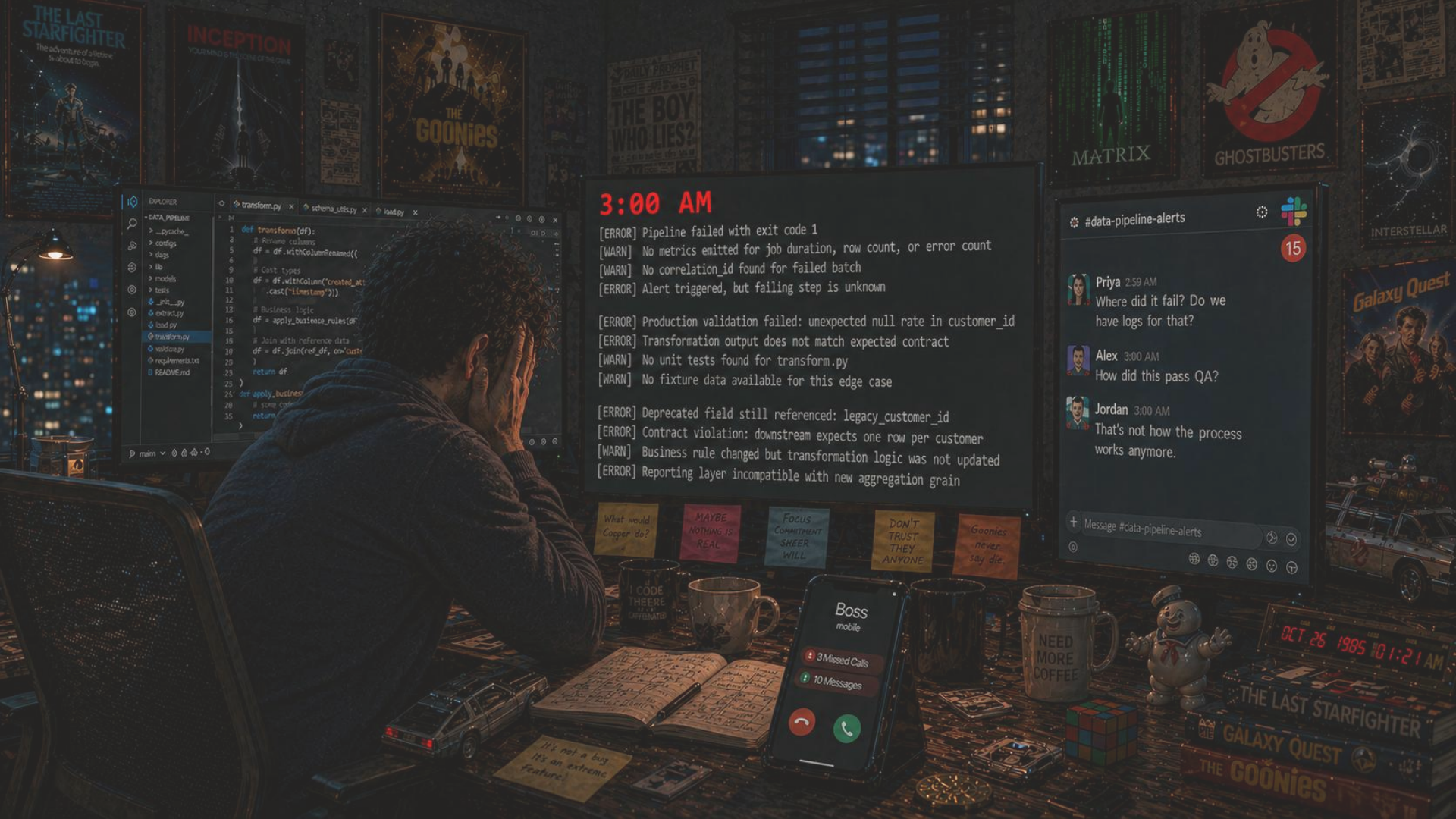
The database will come back up mid-load. The orchestrator will retry on timeout. Someone will ask for a backfill. In a real fight, you don't get one clean exchange — plan for all it.

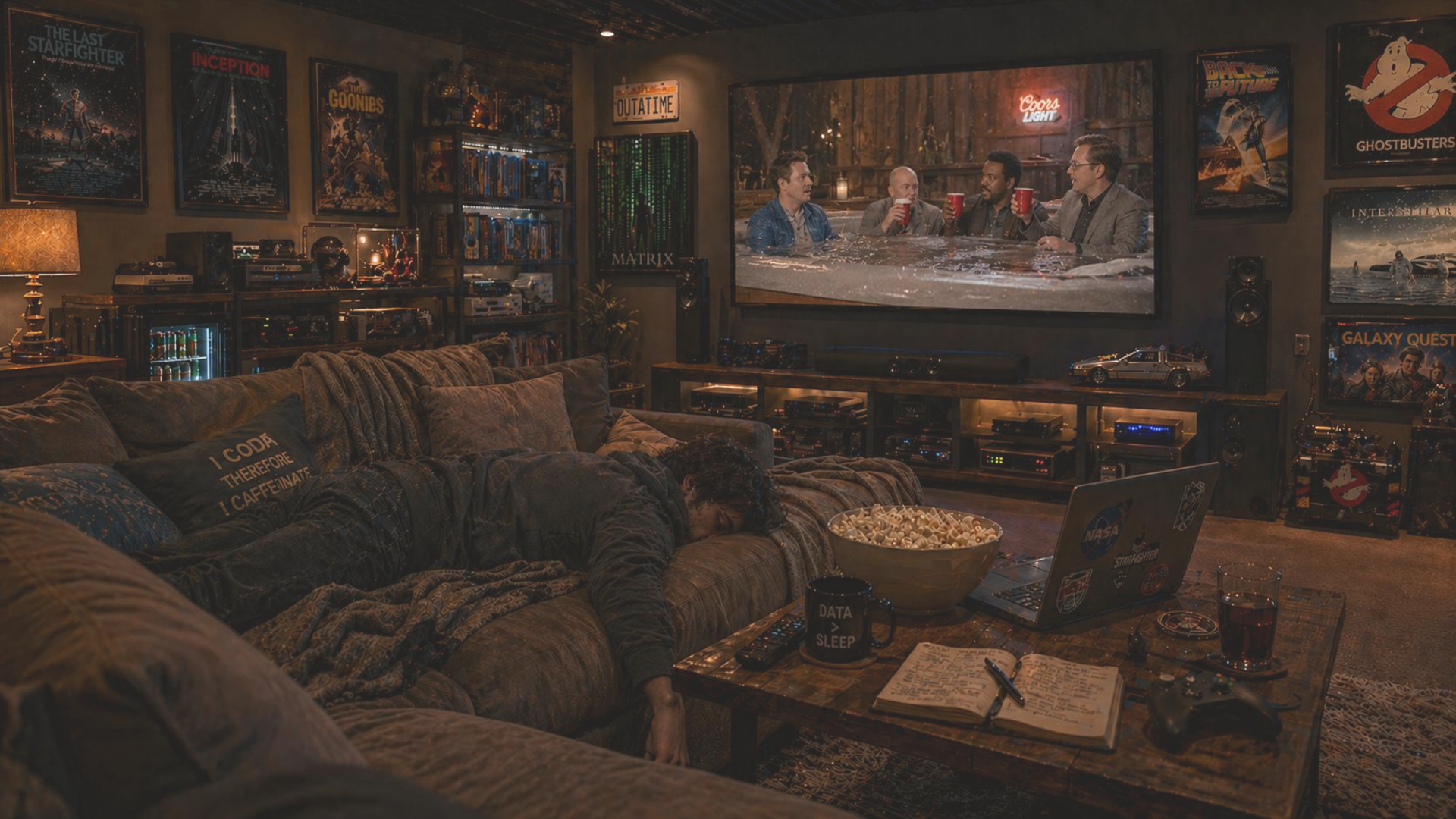
THE FIX

- Deterministic keys
- Upsert over insert
- Safe reprocessing windows
- Ask yourself, is this safe to retry?

"If running it twice breaks it, it was already broken."

THE PAST HAS CHANGED!


3:00 AM
[ERROR] Pipeline failed with exit code 1
[WARN] No metrics emitted for job duration, row count, or error count
[WARN] No correlation_id found for failed batch
[ERROR] Alert triggered, but failing step is unknown
[ERROR] Production validation failed: unexpected null rate in customer_id
[ERROR] Transformation output does not match expected contract
[WARN] No unit tests found for transform.py
[WARN] No fixture data available for this edge case
[ERROR] Deprecated field still referenced: legacy_customer_id
[ERROR] Contract violation: downstream expects one row per customer
[WARN] Business rule changed but transformation logic was not updated
[ERROR] Reporting layer incompatible with new aggregation grain
#data-pipeline-alerts
15
Priya 2:59 AM
Where did it fail? Do we have logs for that?
Alex 3:00 AM
How did this pass QA?
Jordan 3:00 AM
That's not how the process works anymore.
Message #data-pipeline-alerts
Boss
mobile
3 Missed Calls
10 Messages
What would Cooper do?
MAYBE NOTHING IS REAL
Focus Commitment SHEER WILL
DON'T TRUST THEY ANYONE
Goonies never say die.
NEED MORE COFFEE
It's not a bug it's an extreme feature.
OCT 26 1985 01:21 AM
THE LAST STARFIGHTER
GALAXY QUEST
THE GOONIES


THE LAST STARFIGHTER
INCEPTION
THE GOONIES
OUTATIME
MATRIX
Coors LIGHT
BACK TO THE FUTURE
GHOSTBUSTERS
INTERSTELLAR
GALAXY QUEST
I CODA THEREFORE I CAFFEINATE
NASA
STARFIGHTER
DATA > SLEEP

WHAT IF…?

LESSON 03

OBSERVABILITY

Don't charge him blindly. You've got to listen. Listen.

BRUCE'S MISTAKE

6 months ago: Bruce got the data flowing and called it done. No logging beyond basic errors. No metrics. No alerts. The pipeline ran — and that was enough.

Today: The pipeline failed silently for three days. No alerts fired. No dashboards moved. The business found out from a customer complaint about stale data.

Result: Three days of bad data downstream. A frantic audit to determine what was affected. A 3am incident that was completely invisible until it wasn't.

THE LESSON

"The warrior who fights without information fights twice."

Bruce's pipeline was running blind. He knew whether the job finished — not whether the data was right.

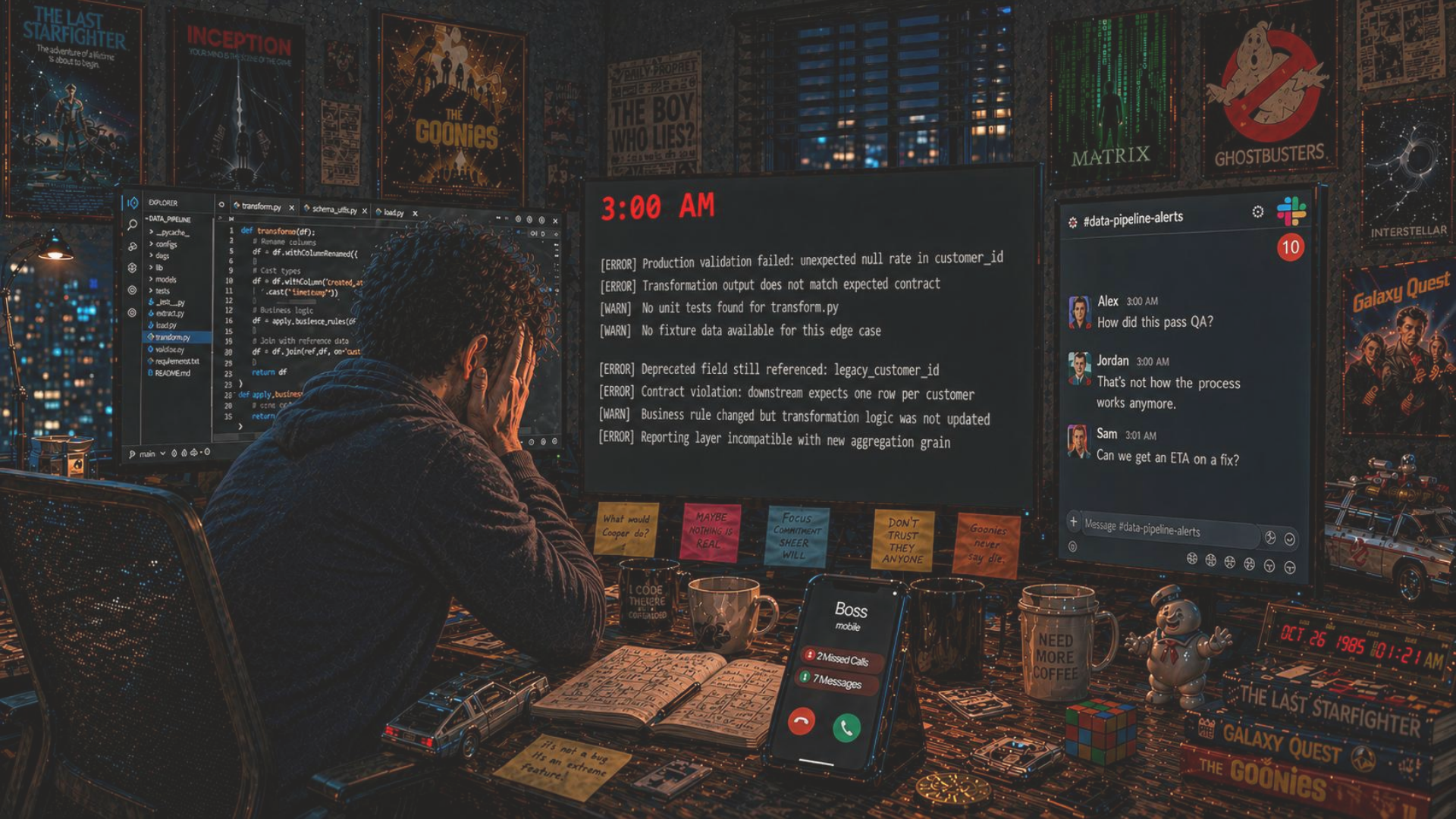
A pipeline that finishes is not the same as a pipeline that works.

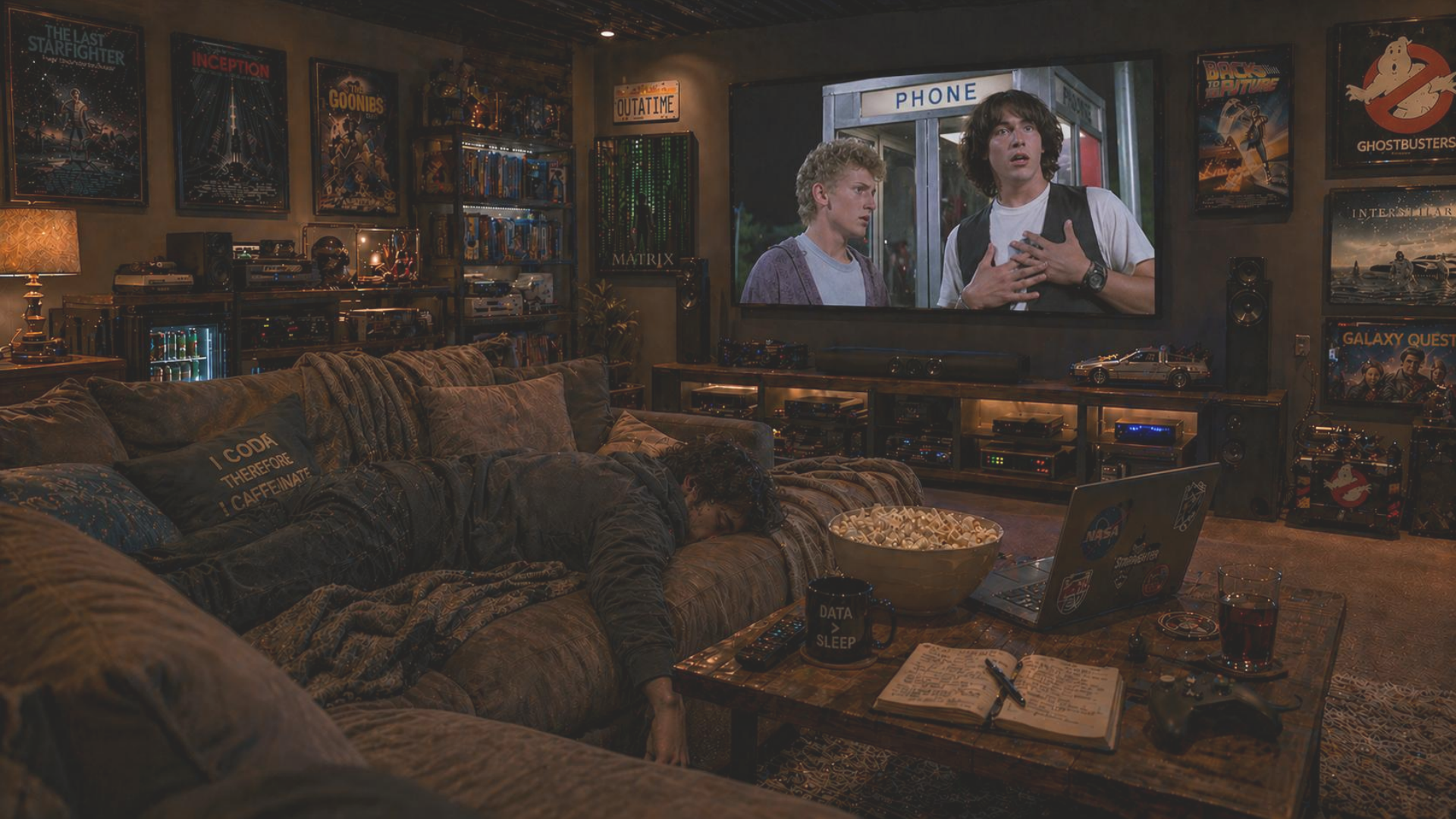
THE FIX

- Row counts at every stage
- Null rate tracking
- Latency monitoring
- Alert on anomalies, not just exceptions

"Hope is not a monitoring strategy."

THE PAST HAS CHANGED!


THE LAST STARFIGHTER
The adventure of a lifetime is about to begin.
INCEPTION
YOUR MIND IS THE SCENE OF THE CRIME
THE GOONIES
THE BOY WHO LIES?
MATRIX
GHOSTBUSTERS
INTERSTELLAR
Galaxy Quest
EXPLORER
DATA_PIPELINE
__pycache__
configs
dags
lib
models
tests
_init__.py
extract.py
load.py
transform.py
validate.py
requirements.txt
README.md
transform.py
schema_utils.py
load.py
def transform(df):
# Rename columns
df = df.withColumnRenamed({
# Cast types
df = df.withColumn("created_at
.cast("timestamp"))
# Business logic
df = apply_business_rules(df
# Join with reference data
df = df.join(ref,df, on="cust
return df
def apply_business
return
main
3:00 AM
[ERROR] Production validation failed: unexpected null rate in customer_id
[ERROR] Transformation output does not match expected contract
[WARN] No unit tests found for transform.py
[WARN] No fixture data available for this edge case
[ERROR] Deprecated field still referenced: legacy_customer_id
[ERROR] Contract violation: downstream expects one row per customer
[WARN] Business rule changed but transformation logic was not updated
[ERROR] Reporting layer incompatible with new aggregation grain
#data-pipeline-alerts
10
Alex 3:00 AM
How did this pass QA?
Jordan 3:00 AM
That's not how the process works anymore.
Sam 3:01 AM
Can we get an ETA on a fix?
Message #data-pipeline-alerts
What would Cooper do?
MAYBE NOTHING IS REAL
FOCUS COMMITMENT SHEER WILL
DON'T TRUST THEY ANYONE
Goonies never say die.
I CODE THERE FOR CAFFEINE
Boss
mobile
2 Missed Calls
7 Messages
NEED MORE COFFEE
it's not a bug it's an extreme feature.
OCT 26 1985 01:21 AM
THE LAST STARFIGHTER
GALAXY QUEST
THE GOONIES


THE LAST STARFIGHTER
INCEPTION
THE GOONIBS
OUTATIME
PHONE
MATRIX
BACK TO THE FUTURE
GHOSTBUSTERS
INTERSTELLAR
GALAXY QUEST
I CODA THEREFORE I CAFFEINATE
NASA
DATA > SLEEP

WHAT IF…?

LESSON 04

TESTABILITY

"What was that? An exhibition?"

BRUCE'S MISTAKE

6 months ago: Bruce built the pipeline and tested it in production. It worked. Data came in, data went out. Looked right. He shipped it.

Today: A bug was found — records with null values in a key field were being silently dropped. The only way to fix it was to run the corrected version in production and watch what happened.

Result: Every change is a live gamble. Every fix introduces new risk. Nobody has confidence in anything because nothing has ever been tested outside of the real environment.

THE LESSON

"I fear not the man who has practiced 10,000 kicks once, but the man who has practiced one kick 10,000 times." — Bruce Lee

You don't find out if a technique works in a real fight. You already know — because you ran it a thousand times in the gym.

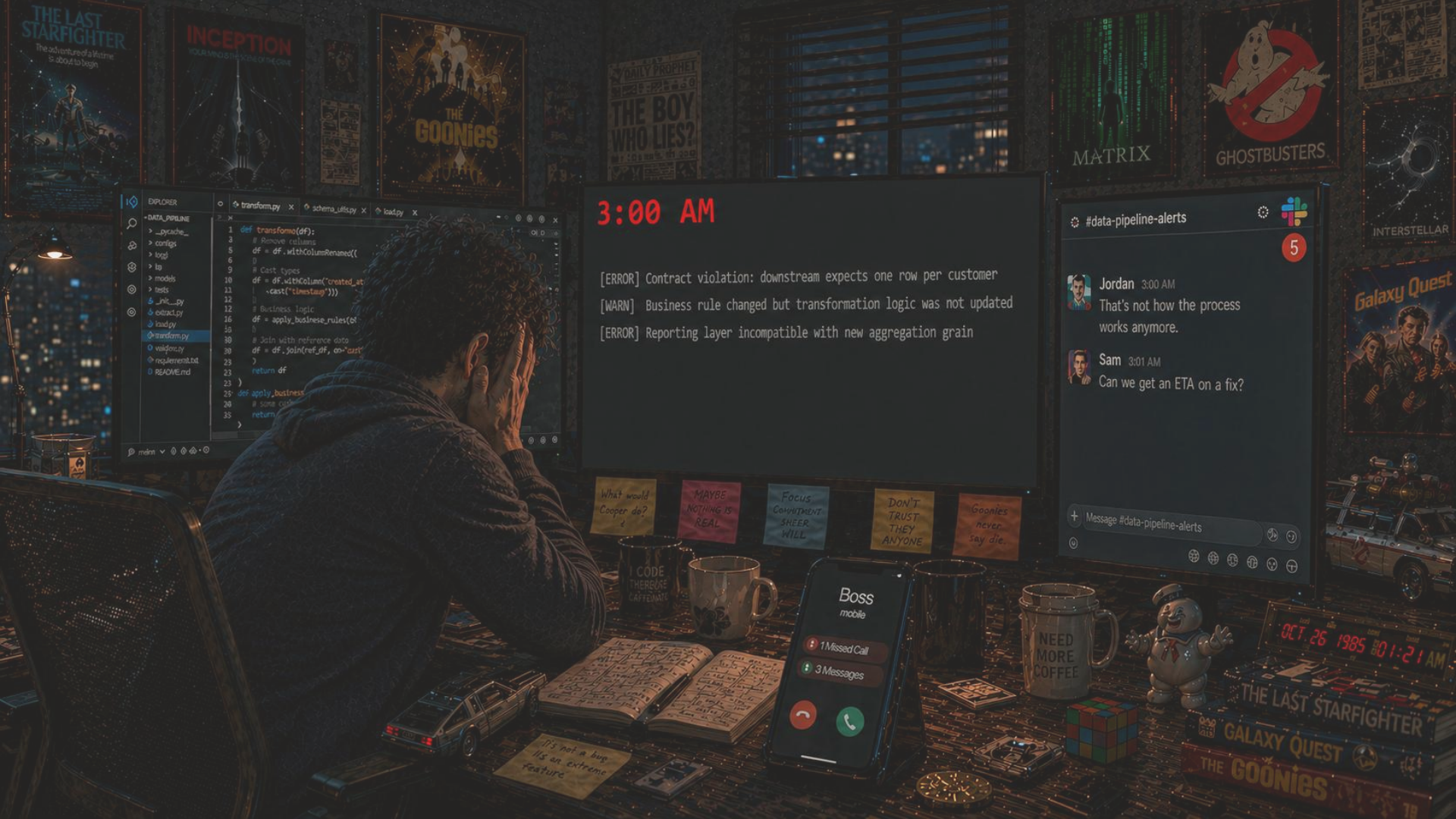
Your pipeline should work the same way. If you've never run it outside of production, you haven't trained at all.

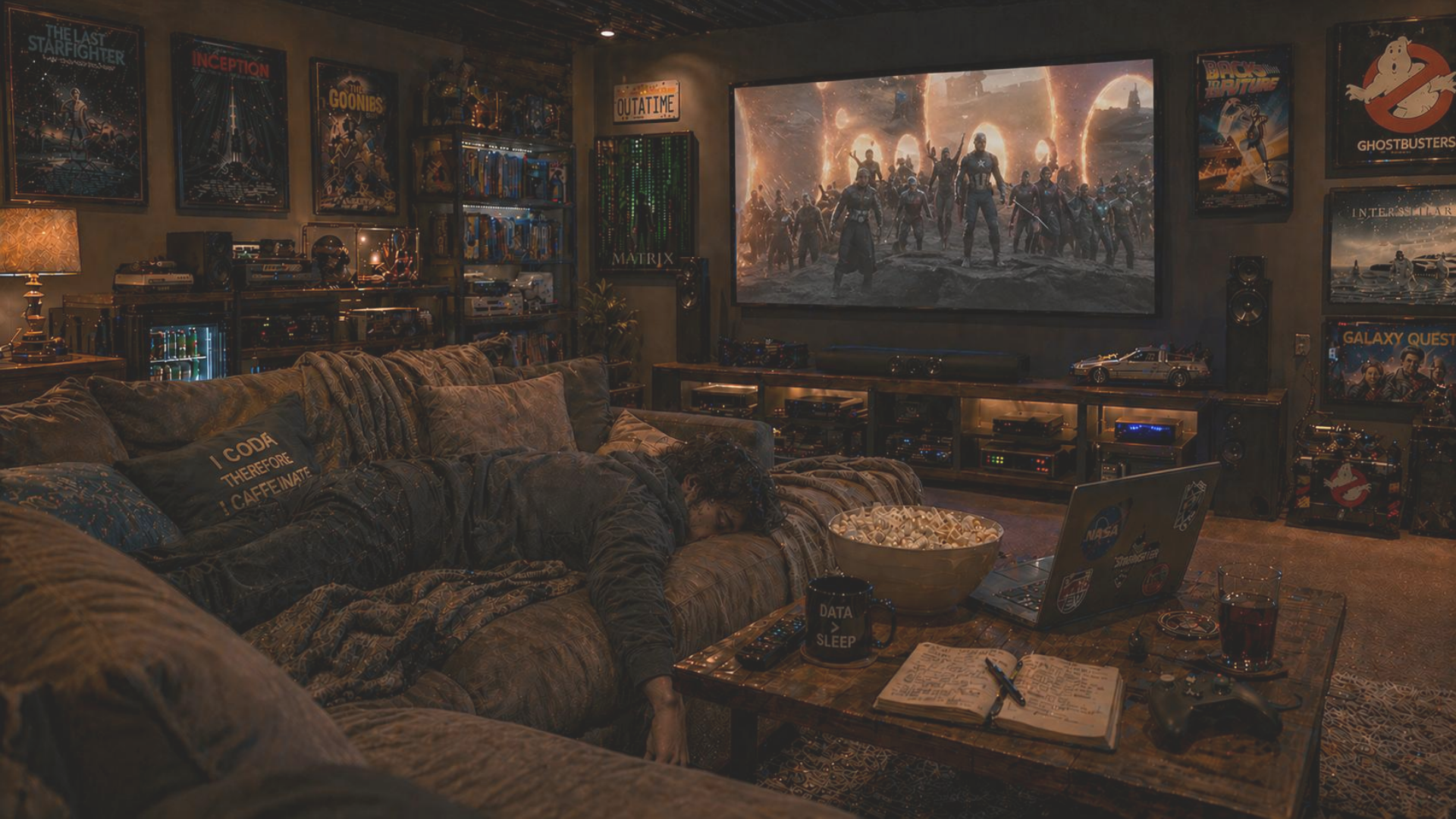
THE FIX

- Pure transformation functions
- Parameterized inputs
- Fixture-based unit tests
- Test without consequences

"Train like you would fight. Don't fight to find out if you trained."

THE PAST HAS CHANGED!


THE LAST STARFIGHTER
INCEPTION
THE GOONIES
DAILY PROPHET
THE BOY WHO LIES?
MATRIX
GHOSTBUSTERS
INTERSTELLAR
Galaxy Quest
EXPLORER
transform.py
schema_utils.py
load.py
DATA_PIPELINE
3:00 AM
[ERROR] Contract violation: downstream expects one row per customer
[WARN] Business rule changed but transformation logic was not updated
[ERROR] Reporting layer incompatible with new aggregation grain
#data-pipeline-alerts
5
Jordan 3:00 AM
That's not how the process works anymore.
Sam 3:01 AM
Can we get an ETA on a fix?
Message #data-pipeline-alerts
What would Cooper do?
MAYBE NOTHING IS REAL
Focus Commitment SHEER WILL
DON'T TRUST THEY ANYONE
Goonies never say die.
I CODE THEREFORE CAFFEINATED
Boss
mobile
1 Missed Call
3 Messages
NEED MORE COFFEE
It's not a bug it's an extreme feature
OCT 26 1985 01:21 AM
THE LAST STARFIGHTER
GALAXY QUEST
THE GOONIES


THE LAST STARFIGHTER
INCEPTION
THE GOONIES
OUTATIME
MATRIX
BACK TO THE FUTURE
GHOSTBUSTERS
INTERSTELLAR
GALAXY QUEST
I CODE THEREFORE I CAFFEINATE
DATA > SLEEP
NASA

WHAT IF…?

LESSON 05

DESIGN FOR CHANGE

"Be water, my friend"

BRUCE'S MISTAKE

6 months ago: Bruce's pipeline was tight, clean, and exactly right for the problem in front of him. It handled the volume, the schema, the delivery cadence. It was built for the fight he knew was coming.

Today: Volume is 100x what it was. Requirements shifted. The source system restructured. The business wants to backfill 18 months of history.

Result: The pipeline didn't bend. It shattered. A full rewrite — not because the original was bad, but because it was built for one specific fight and the fight changed.

THE LESSON

"Notice that the stiffest tree is most easily cracked, while the bamboo or willow survives by bending with the wind."

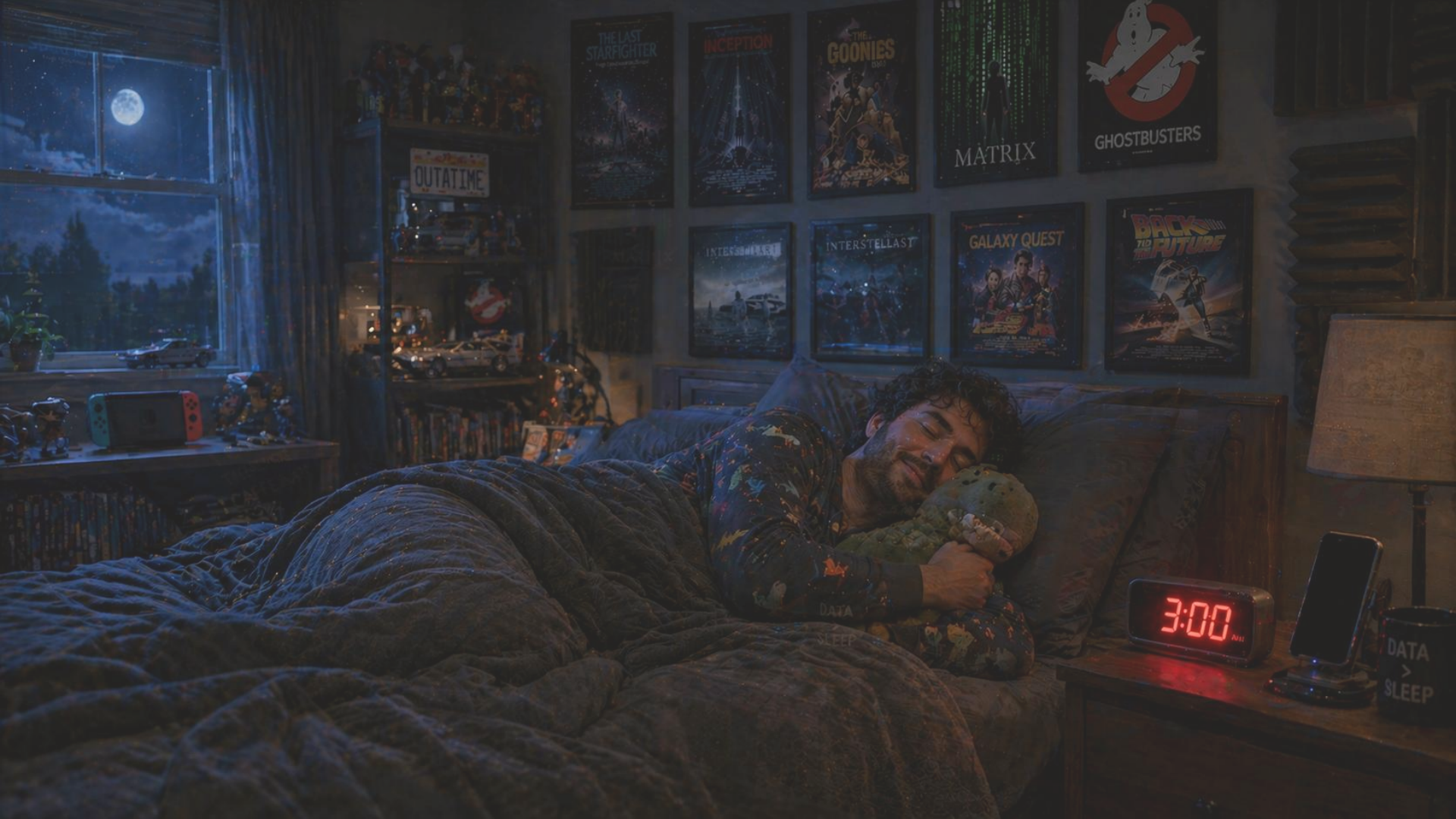
Water doesn't resist the container. It adapts. Bruce's pipeline was concrete — perfectly shaped for the original mold. When the mold changed, the concrete cracked. Build pipelines that bend.

THE FIX

- Modular pipeline stages
- Backfill-friendly design
- Smart partition strategies
- Loose coupling between layers

"Volume is the easy scaling problem. Changing requirements is the real test."

THE PAST HAS CHANGED!


THE LAST STARFIGHTER
INCEPTION
THE GOONIES
MATRIX
GHOSTBUSTERS
OUTATIME
INTERSTELLAST
GALAXY QUEST
BACK TO THE FUTURE
3:00
DATA
SLEEP
DATA
>
SLEEP

THE REMAKE

Bruce Redeemed.

BEFORE

- Silent failures at 3am
- Doubled data on reruns
- Untestable, unable to do any meaningful performance tests or stress tests
- Explodes when schema changes
- Full rewrite when requirements shift

AFTER

- Anomaly alerts catch issues early and fix them or move on not breaking the pipeline
- Idempotent — safe to retry
- Full test suite, no consequences
- Schema evolution handled gracefully
- Modular — bends instead of breaks

BEFORE YOU GO WATCH THE MOVIE

01 Build for schema change.

02 Make it idempotent.

03 Observe everything.

04 Test without consequences.

05 Design for Changes…Be water.

Bruce left before he could fix it.

You're still here.

You can fix it!

Start Small, Do Something

"Go enjoy the movie. You've already seen the real disaster film."

Chris Birie
Software & Data Martial Artist
chris.birie@gmail.com

Feedback

LinkedIn