wave

CONF42

Instrumenting at
10 years per second

Jean-Mark Wright

Wave Financial Inc

wave

About me

JEAN-MARK WRIGHT
Staff Engineer, Observability
Wave Financial Inc

Agenda

01
PROBLEM

02
SOLUTION

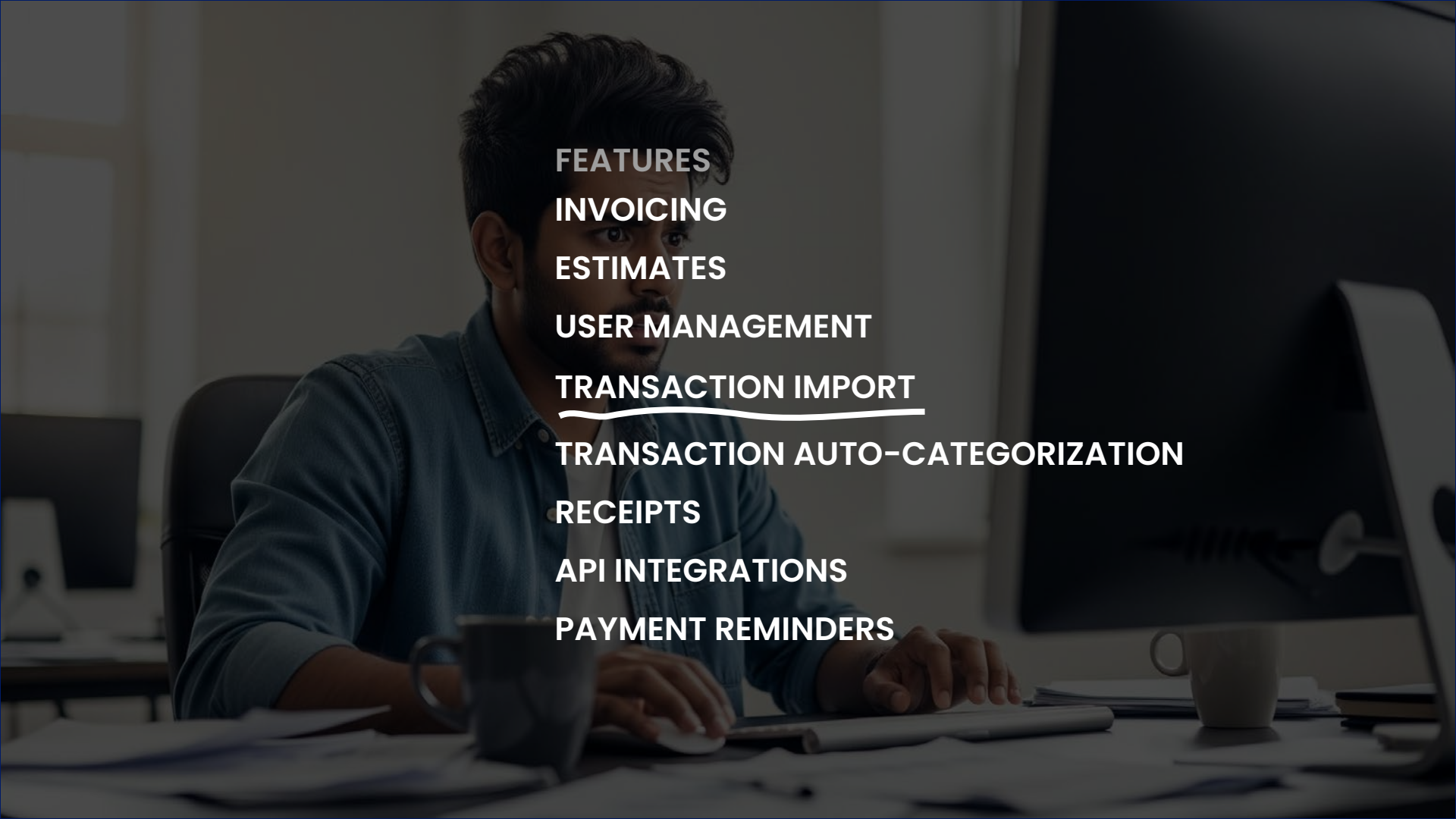
03
HOW

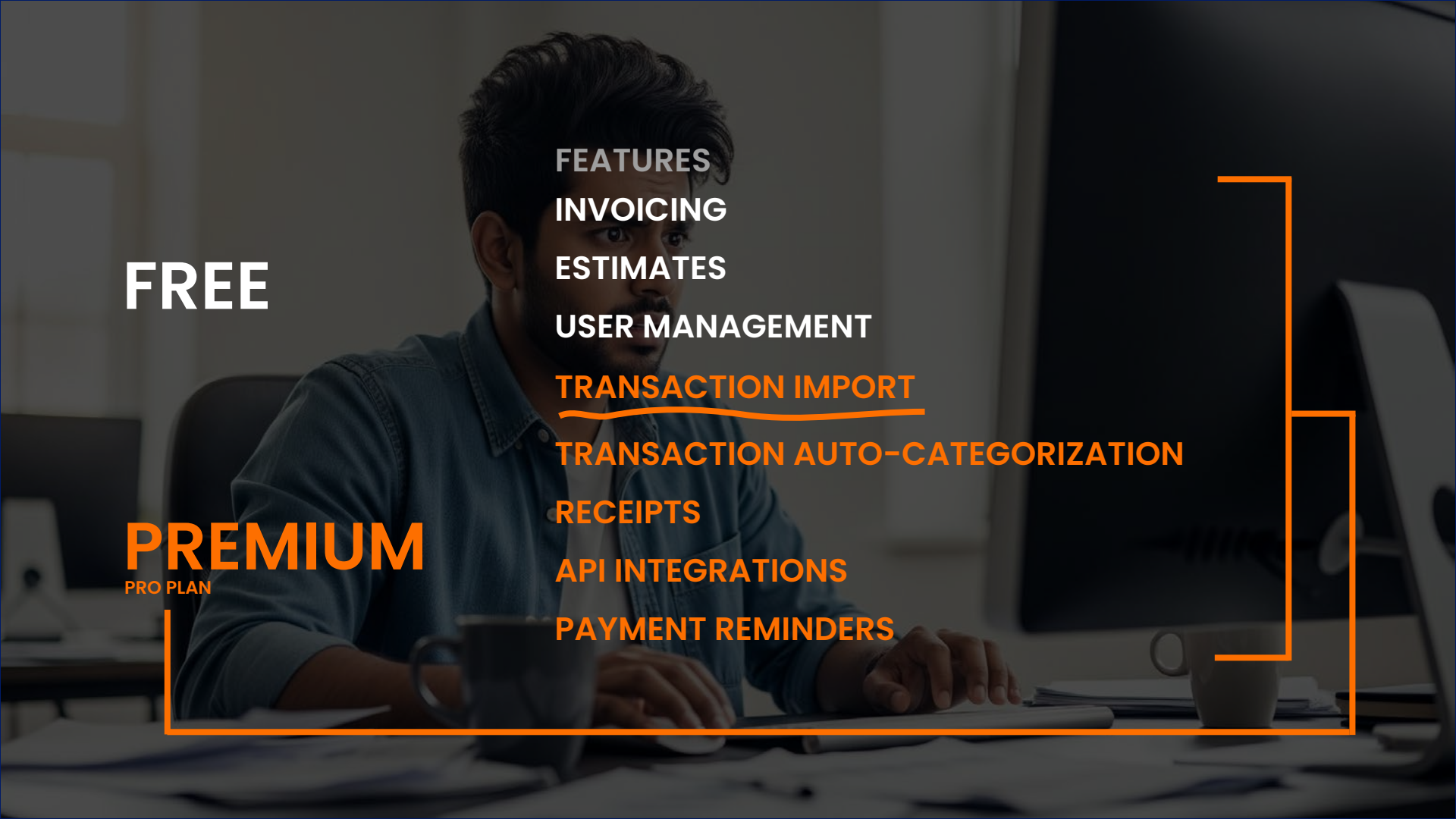
04
NEXT STEPS

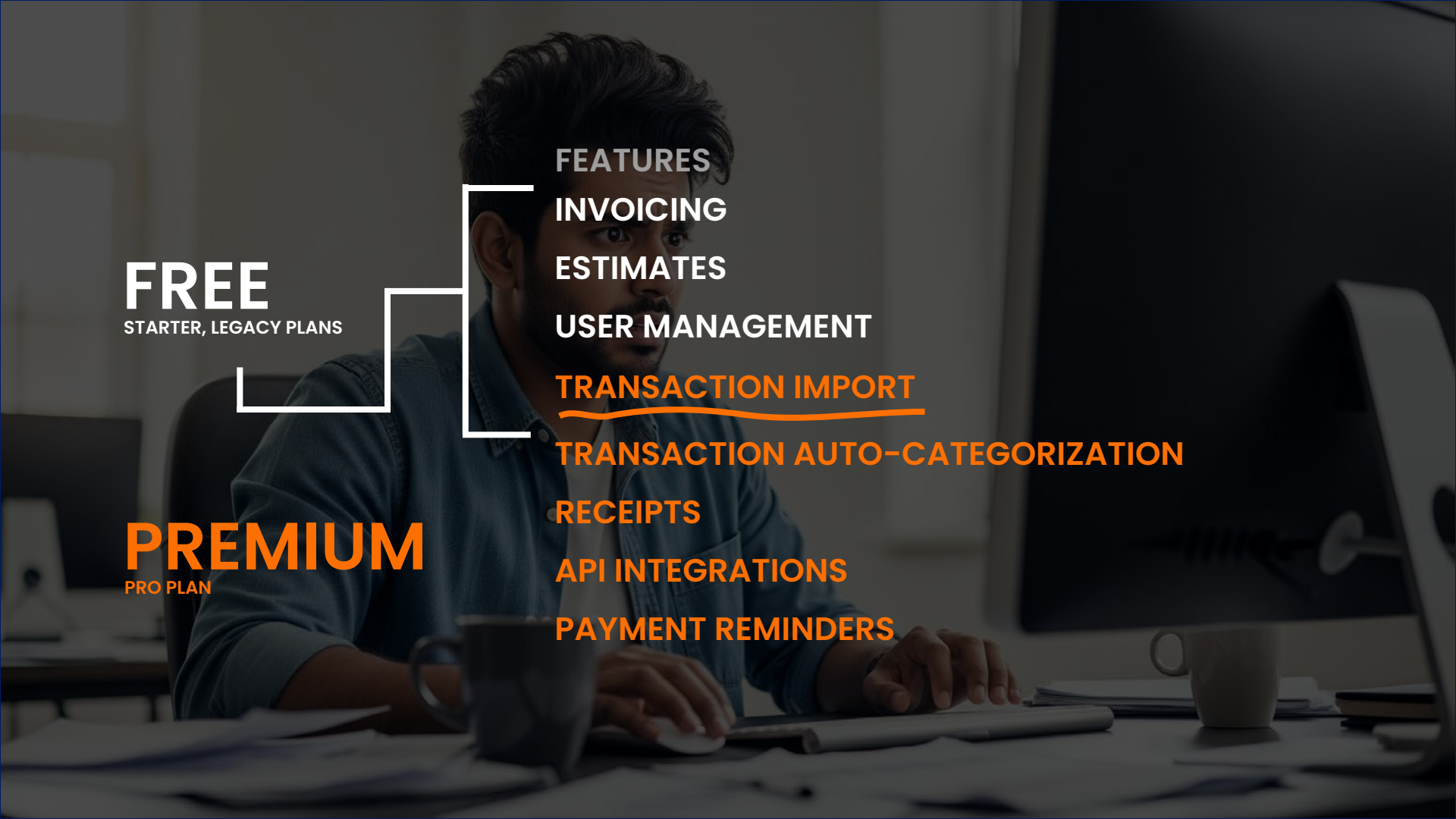

MANUAL INSTRUMENTATION IS POWERFUL
EVERY ENGINEER DESERVES IT
"Debugging a premium leak"

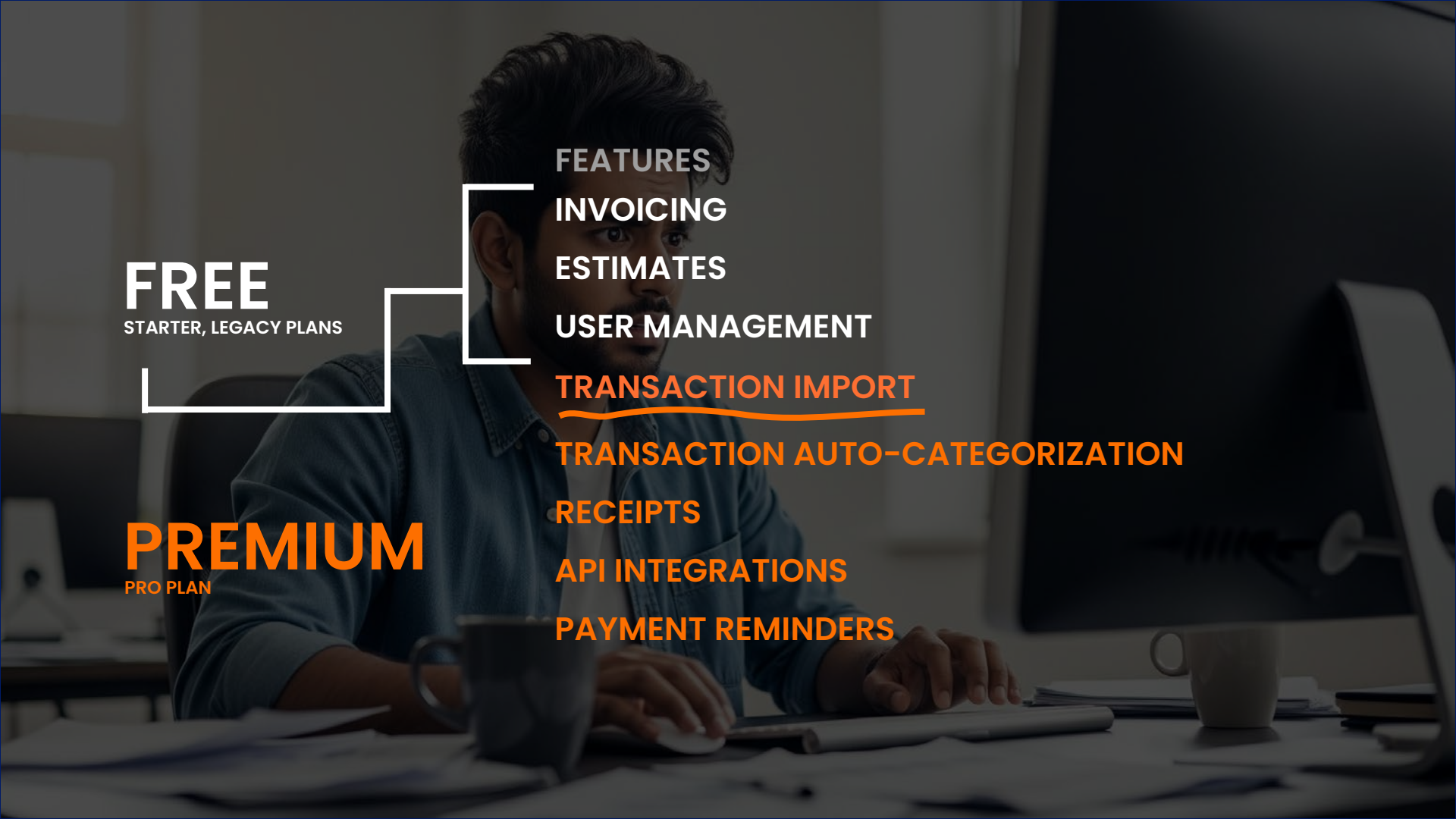

PREMIUM
FREE
FEATURES

Connect your bank & **import your transactions** into Accounting


FEATURES
INVOICING
ESTIMATES
USER MANAGEMENT
TRANSACTION IMPORT
TRANSACTION AUTO-CATEGORIZATION
RECEIPTS
API INTEGRATIONS
PAYMENT REMINDERS


FREE
PREMIUM
PRO PLAN
FEATURES
INVOICING
ESTIMATES
USER MANAGEMENT
TRANSACTION IMPORT
TRANSACTION AUTO-CATEGORIZATION
RECEIPTS
API INTEGRATIONS
PAYMENT REMINDERS


FREE
STARTER, LEGACY PLANS
PREMIUM
PRO PLAN
FEATURES
INVOICING
ESTIMATES
USER MANAGEMENT
TRANSACTION IMPORT
TRANSACTION AUTO-CATEGORIZATION
RECEIPTS
API INTEGRATIONS
PAYMENT REMINDERS


FREE
STARTER, LEGACY PLANS
FEATURES
INVOICING
ESTIMATES
USER MANAGEMENT
TRANSACTION IMPORT
TRANSACTION AUTO-CATEGORIZATION
RECEIPTS
API INTEGRATIONS
PAYMENT REMINDERS
PREMIUM
PRO PLAN

This application was particularly special

AUTO-INSTRUMENTATION METRICS TRACES LOGS

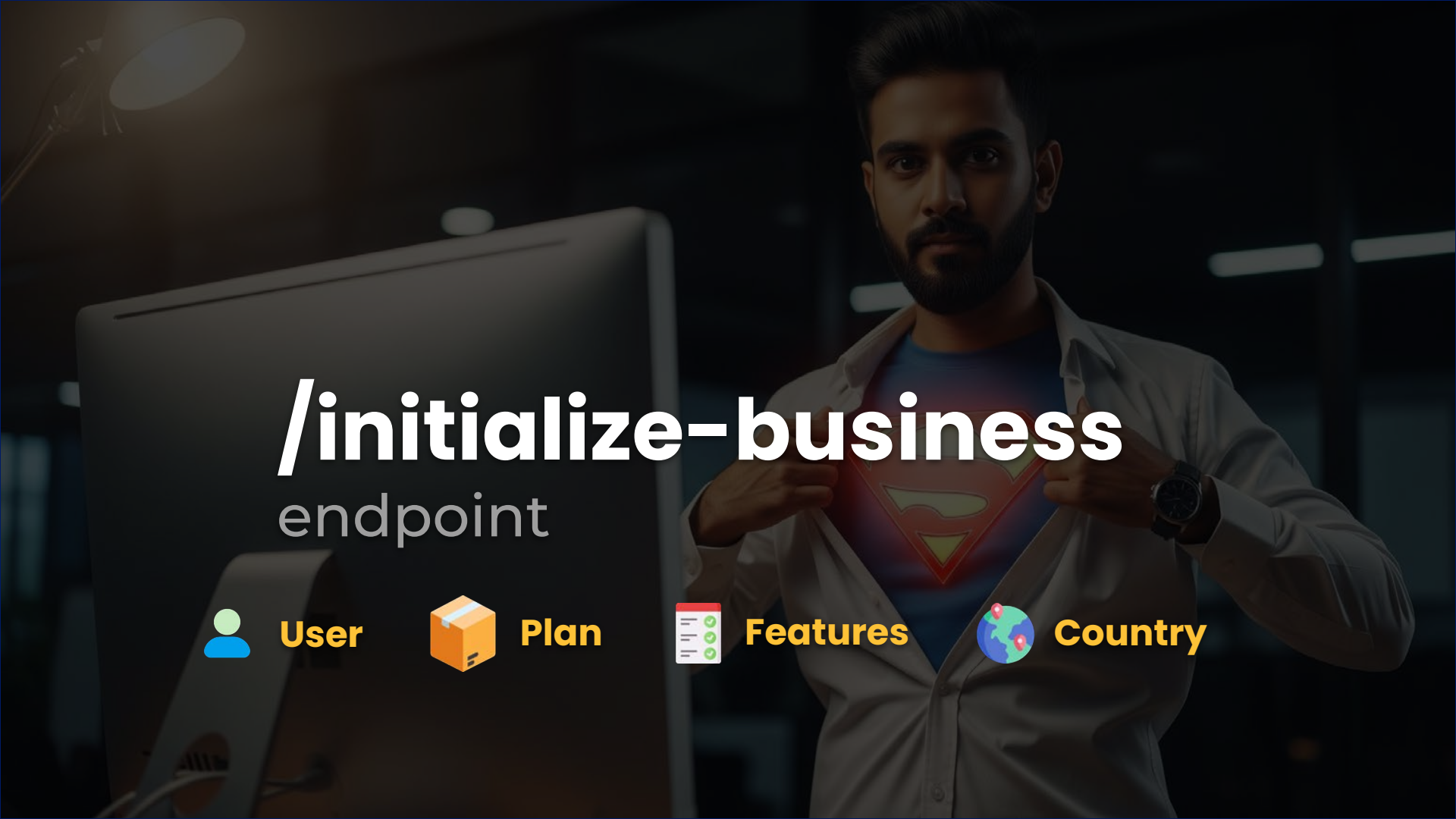

HIGH
QUALITY
MANUAL
INSTRUMENTATION

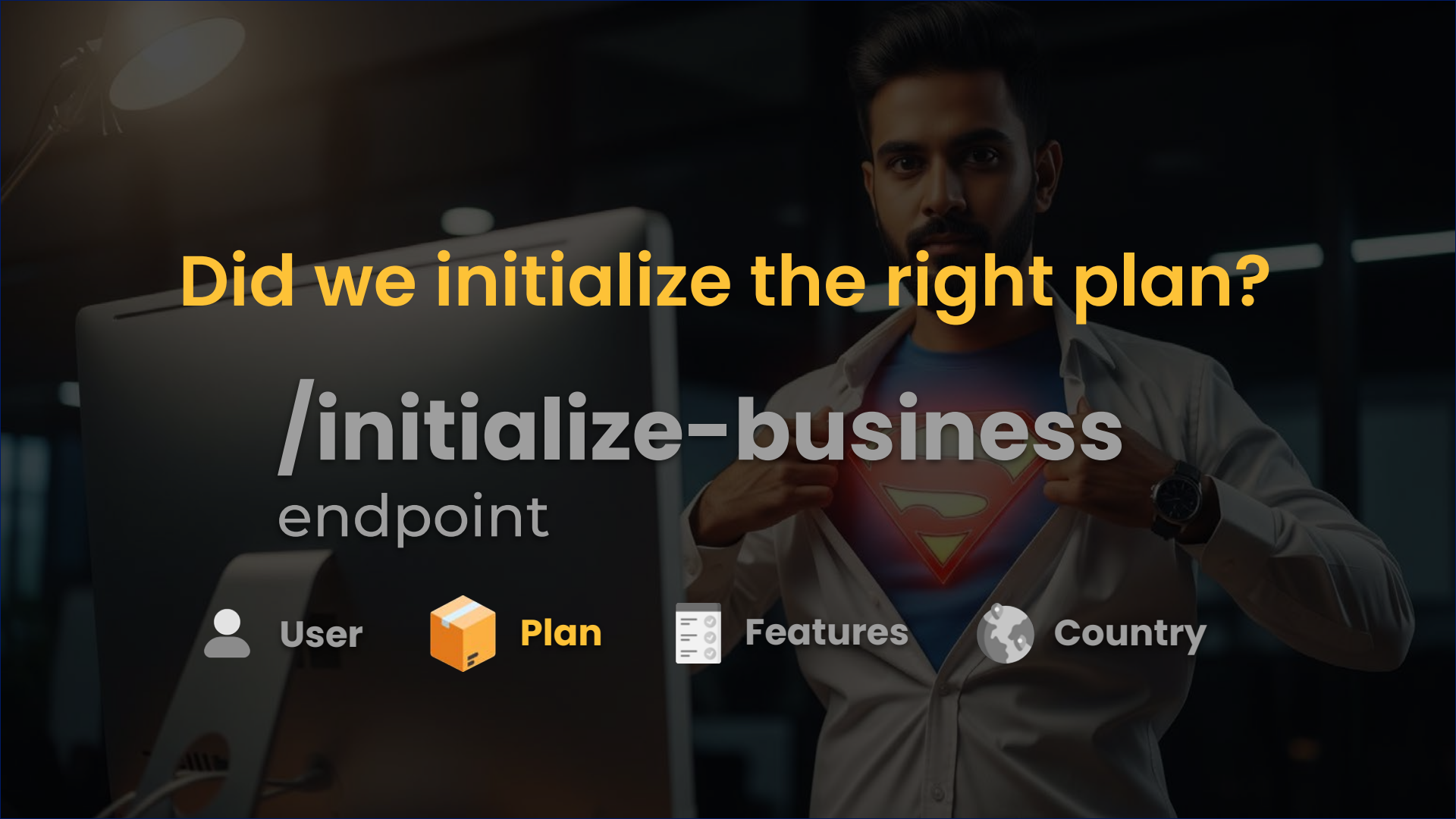

STARTER PLAN
LEGACY PLAN
/initialize-business
endpoint

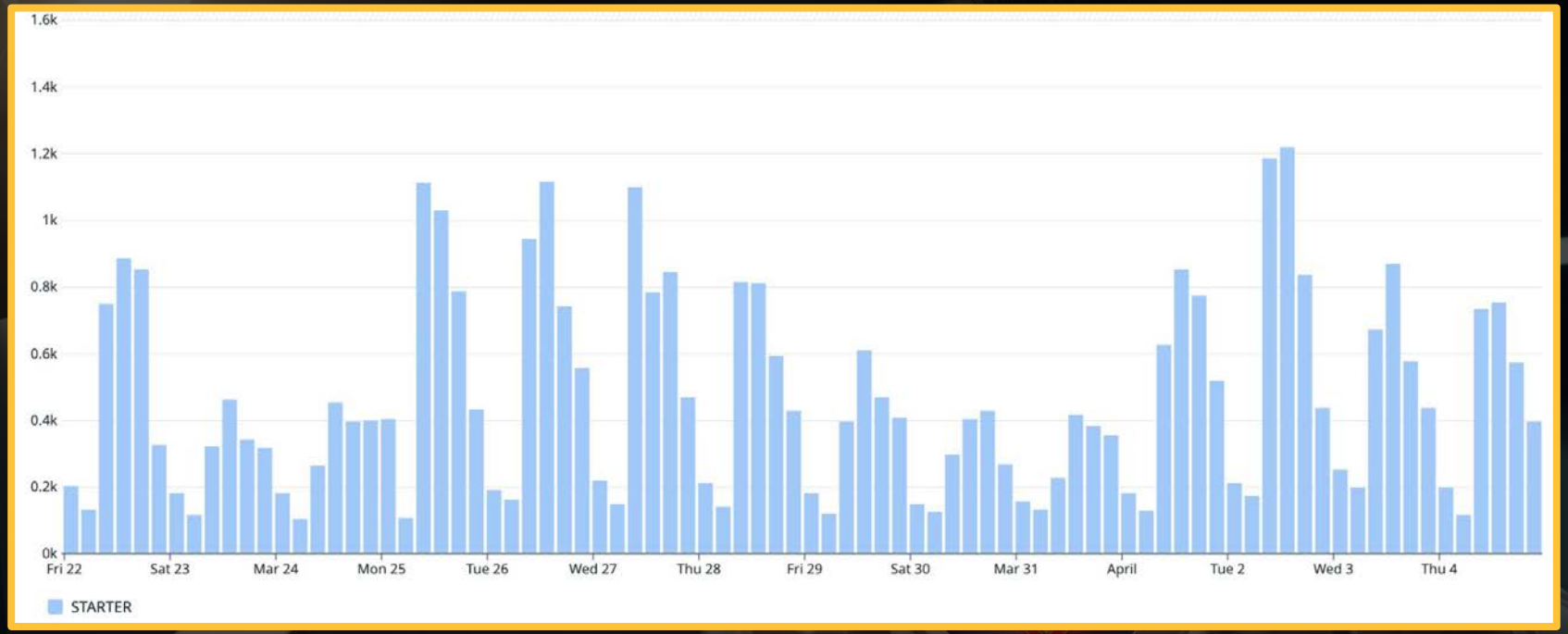

STARTER PLAN
LEGACY PLAN
/initialize-business
endpoint


/initialize-business
endpoint


/initialize-business
endpoint
User

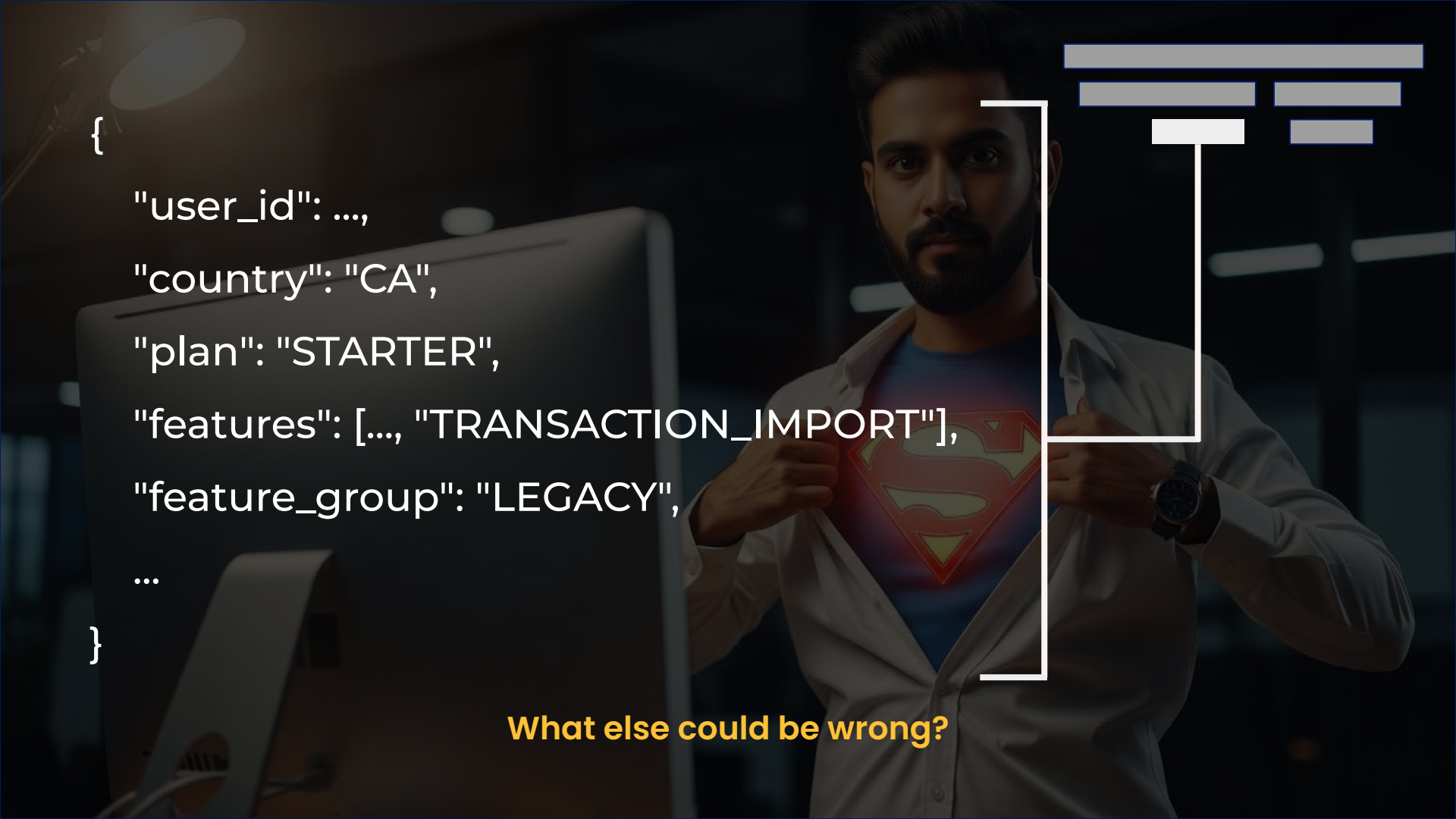

/initialize-business
endpoint
User
Plan
Features
Country

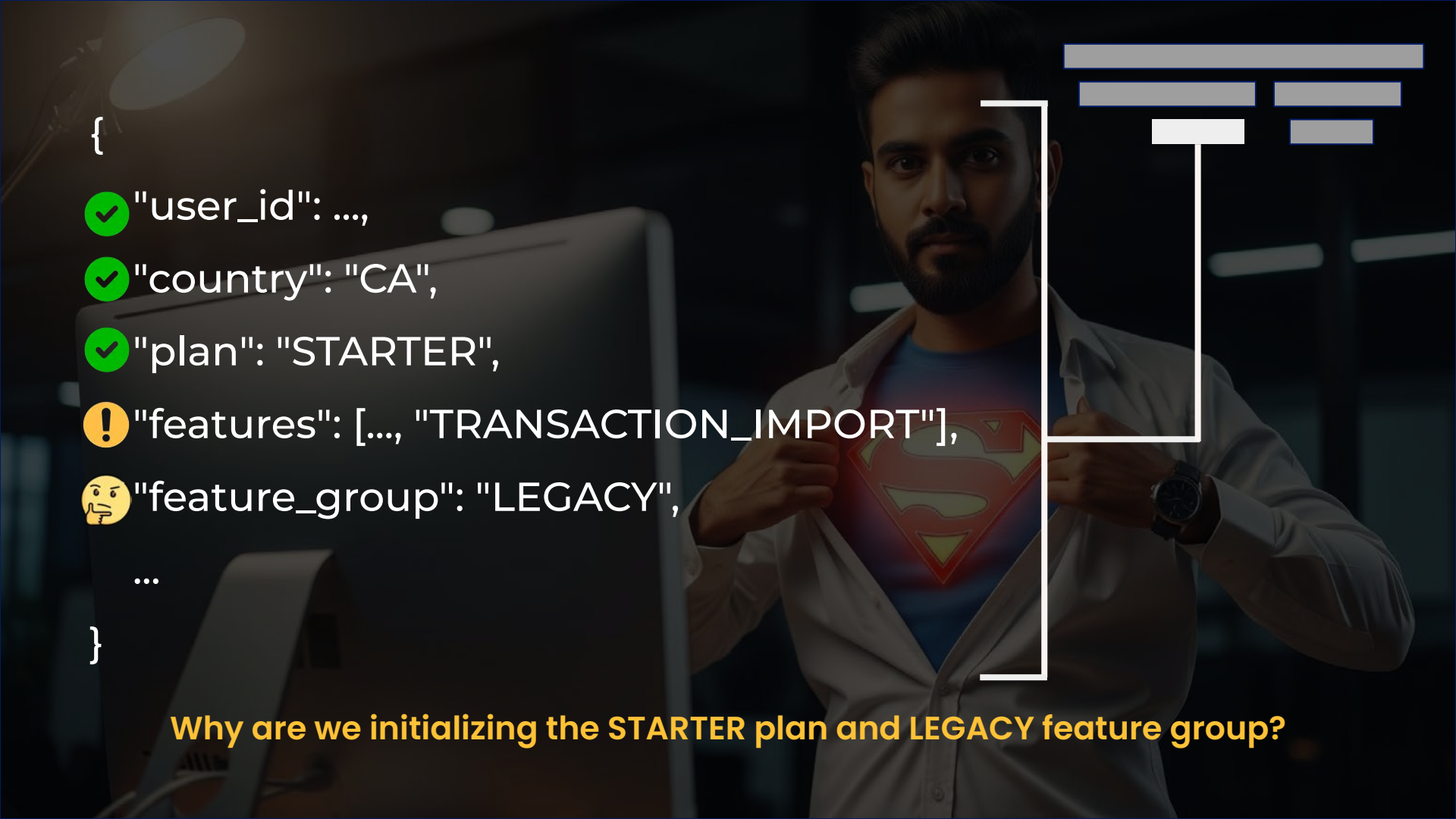

Did we initialize the right plan?
/initialize-business
endpoint
User
Plan
Features
Country

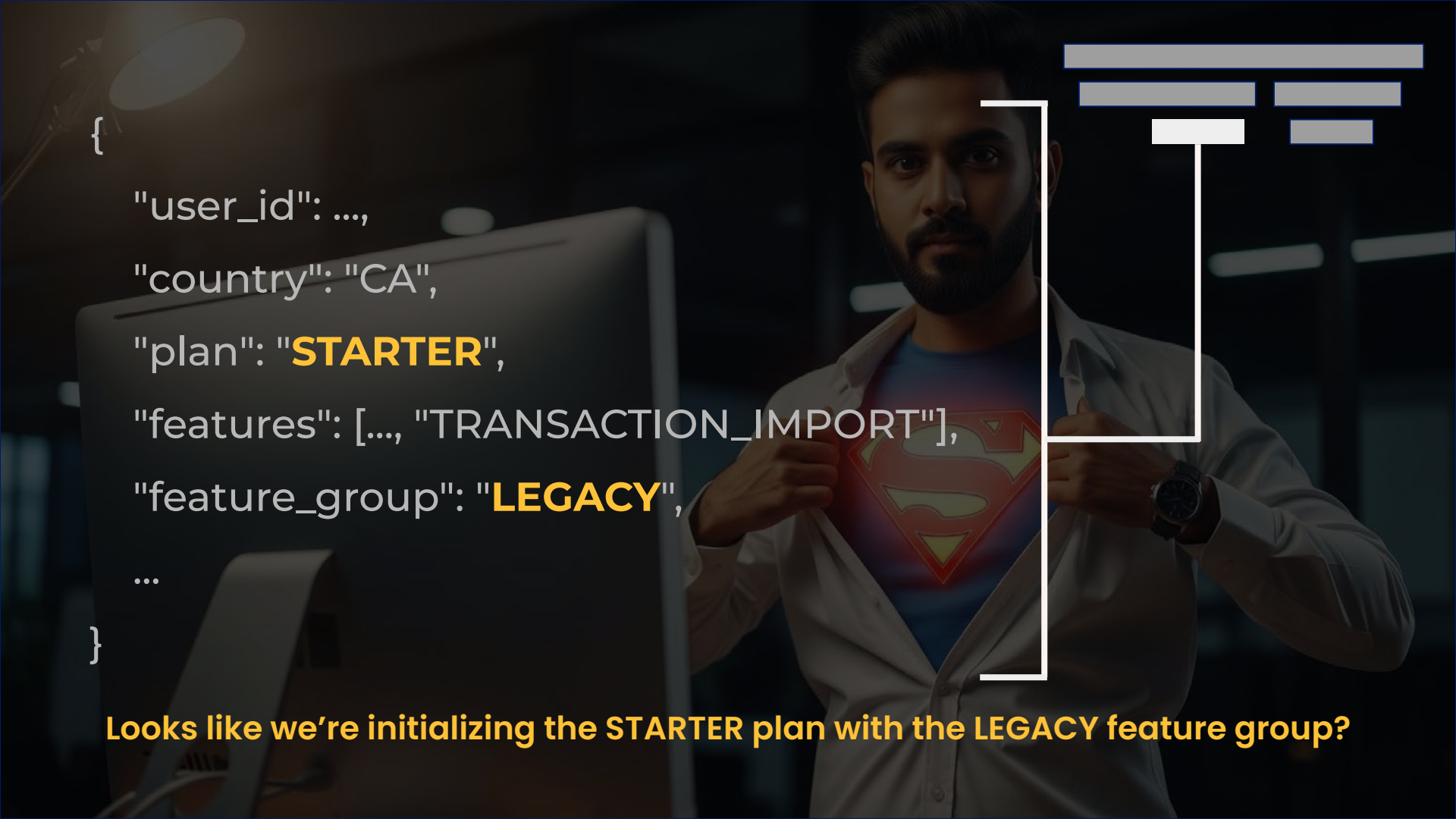


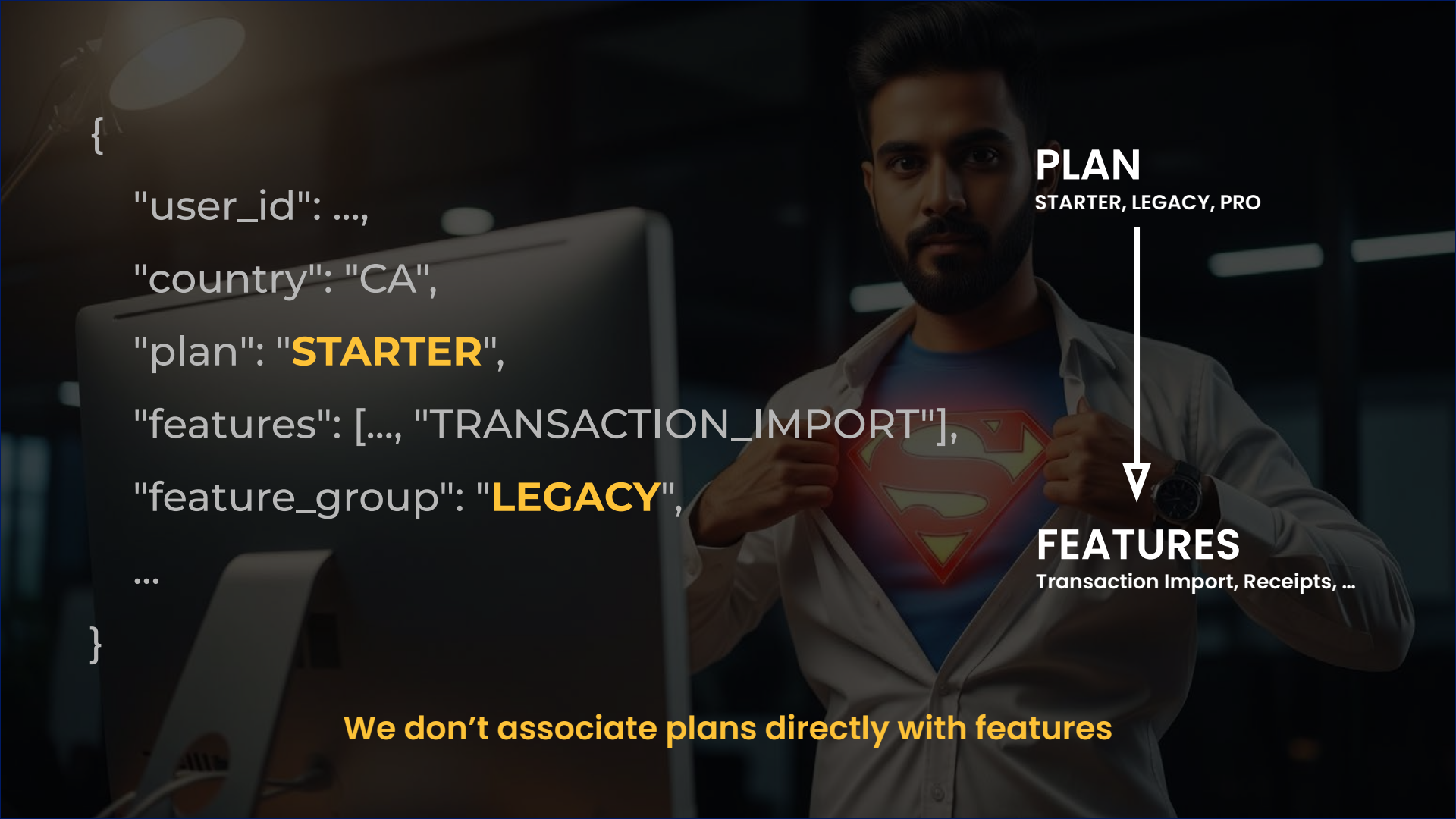
User **Plan** Features Country

Let’s examine initialized plans in recent traces

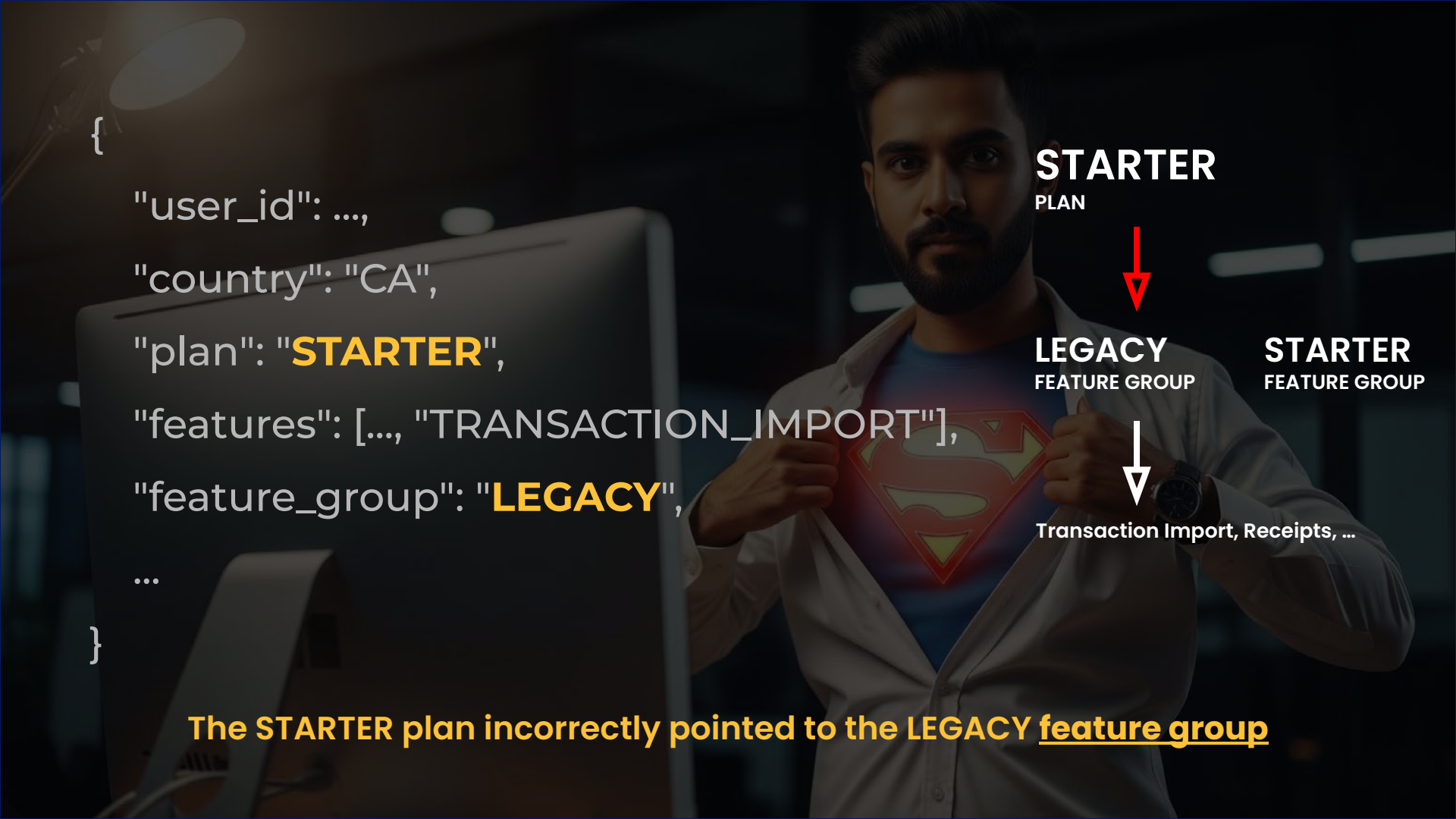

{
"user_id": ...,
"country": "CA",
"plan": "STARTER",
"features": [..., "TRANSACTION_IMPORT"],
"feature_group": "LEGACY",
...
}
What else could be wrong?

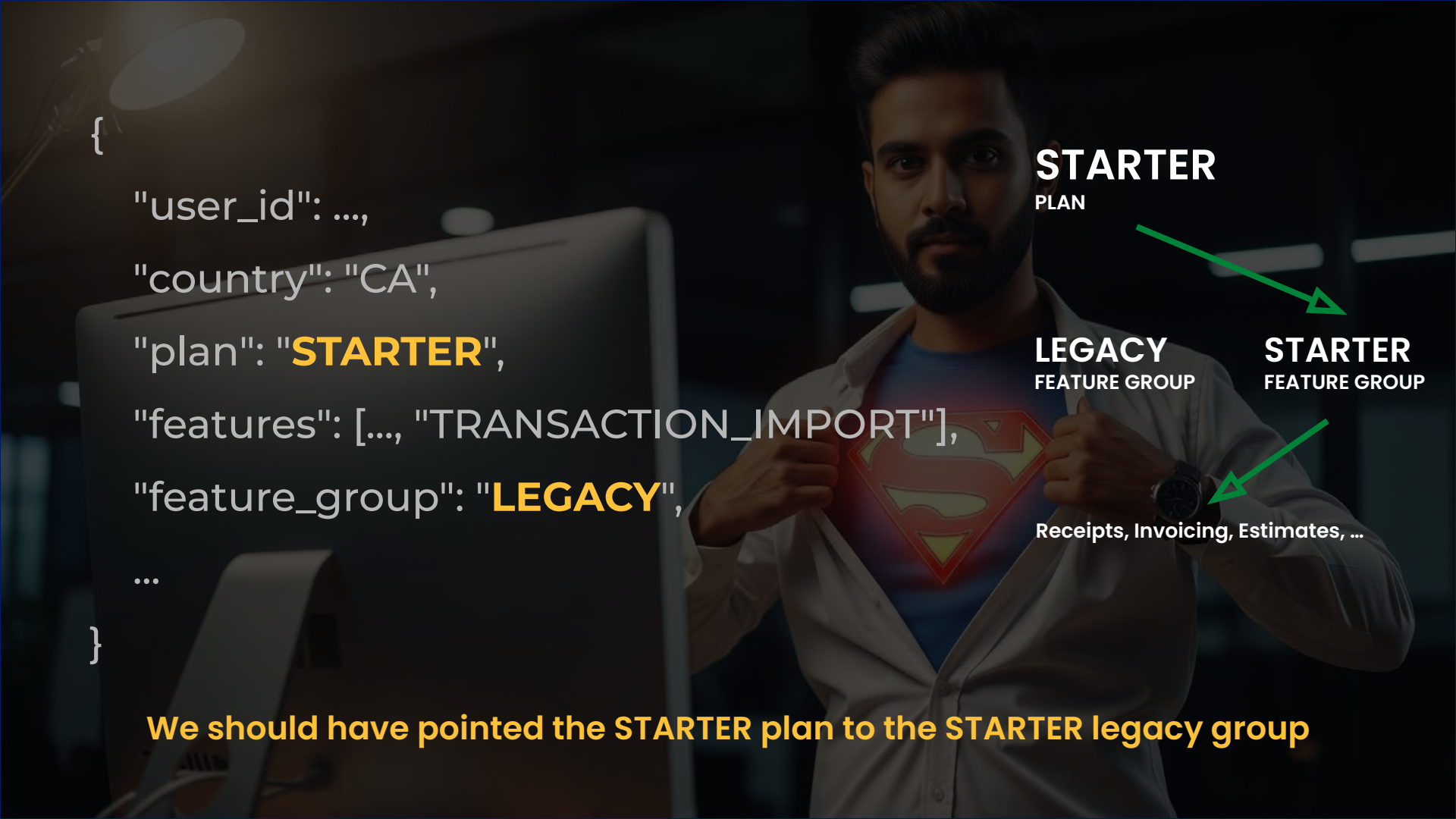

{
"user_id": ...,
"country": "CA",
"plan": "STARTER",
"features": [..., "TRANSACTION_IMPORT"],
"feature_group": "LEGACY",
...
}
Why are we initializing the STARTER plan and LEGACY feature group?

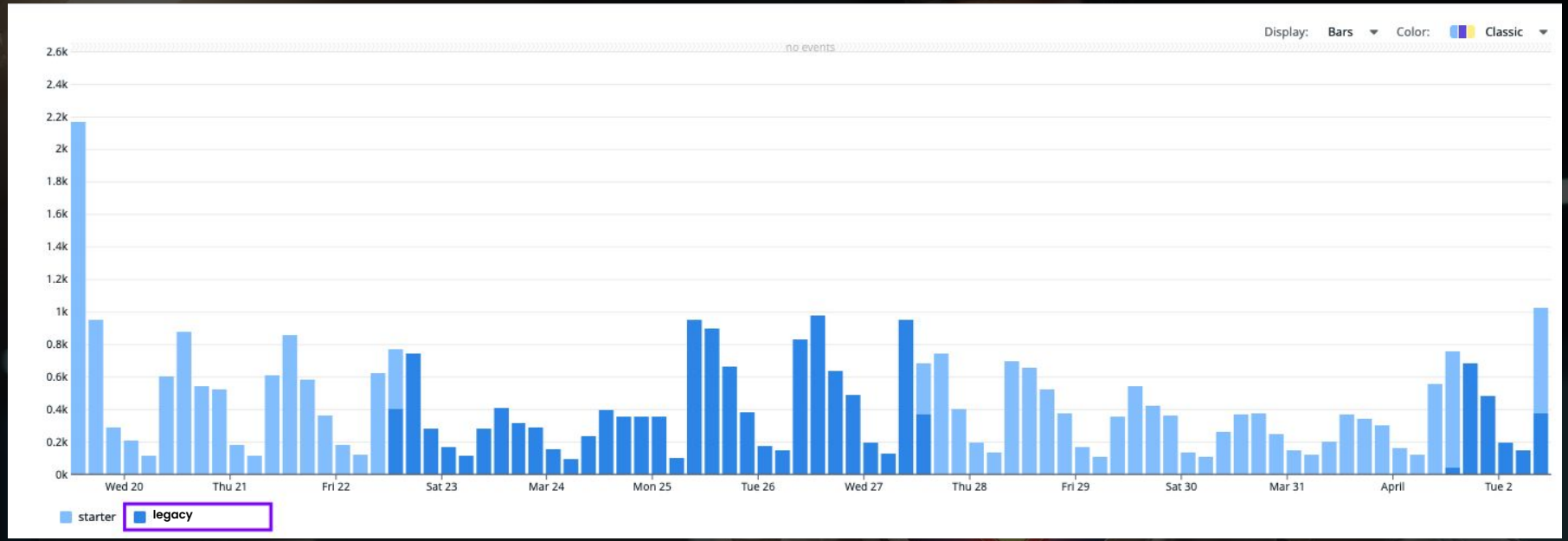

{
"user_id": ...,
"country": "CA",
"plan": "STARTER",
"features": [..., "TRANSACTION_IMPORT"],
"feature_group": "LEGACY",
...
}
Looks like we're initializing the STARTER plan with the LEGACY feature group?


{
"user_id": ...,
"country": "CA",
"plan": "STARTER",
"features": [..., "TRANSACTION_IMPORT"],
"feature_group": "LEGACY",
...
}
PLAN
STARTER, LEGACY, PRO
FEATURES
Transaction Import, Receipts, ...
We don't associate plans directly with features


{
"user_id": ...,
"country": "CA",
"plan": "STARTER",
"features": [..., "TRANSACTION_IMPORT"],
"feature_group": "LEGACY",
...
}
PLAN
STARTER, LEGACY, PRO
FEATURE GROUP
STARTER, LEGACY, PRO
FEATURES
Transaction Import, Receipts, ...
We put features in a group then associate then to a plan


{
"user_id": ...,
"country": "CA",
"plan": "STARTER",
"features": [..., "TRANSACTION_IMPORT"],
"feature_group": "LEGACY",
...
}
STARTER
PLAN
LEGACY
FEATURE GROUP
STARTER
FEATURE GROUP
Transaction Import, Receipts, ...
The STARTER plan incorrectly pointed to the LEGACY feature group


{
"user_id": ...,
"country": "CA",
"plan": "STARTER",
"features": [..., "TRANSACTION_IMPORT"],
"feature_group": "LEGACY",
...
}
STARTER
PLAN
LEGACY
FEATURE GROUP
STARTER
FEATURE GROUP
Receipts, Invoicing, Estimates, ...
We should have pointed the STARTER plan to the STARTER legacy group



Plan Features User Country Version

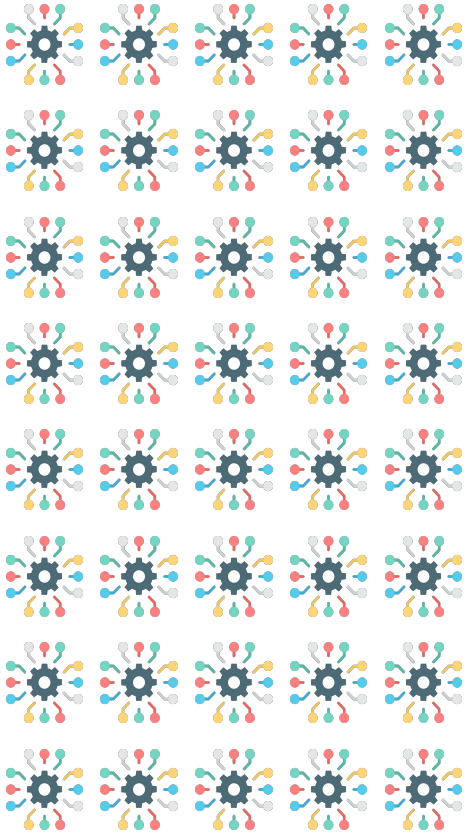
What feature groups are we initializing in our traces?


12 mins
ROOT CAUSE ANALYSIS
3 mins
FIX DATA CORRUPTION
Fixed
"Debugging a premium leak"

THE PROBLEM

Wasn’t that a great outcome?

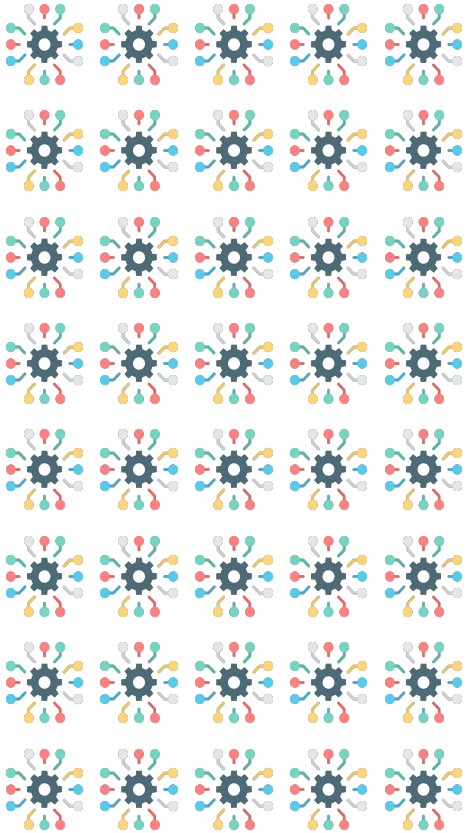
One well-instrumented microservice



40+ additional microservices

~3 months

TIME TO INSTRUMENT

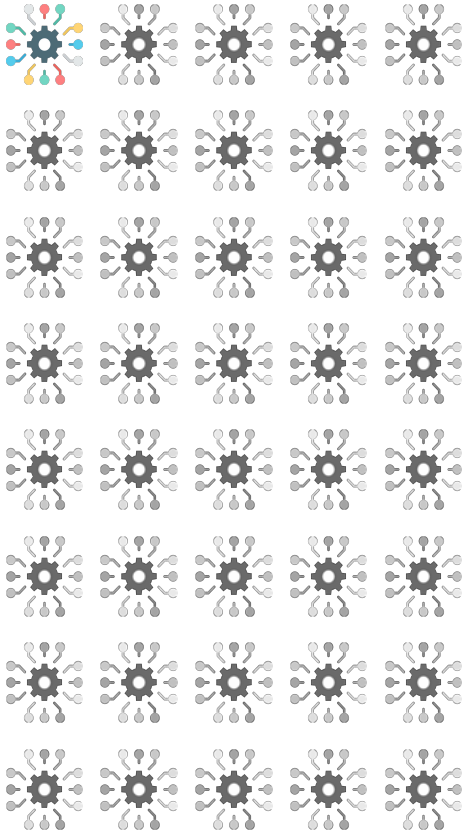


3 months X 40 services = 10 years

TIME TO INSTRUMENT

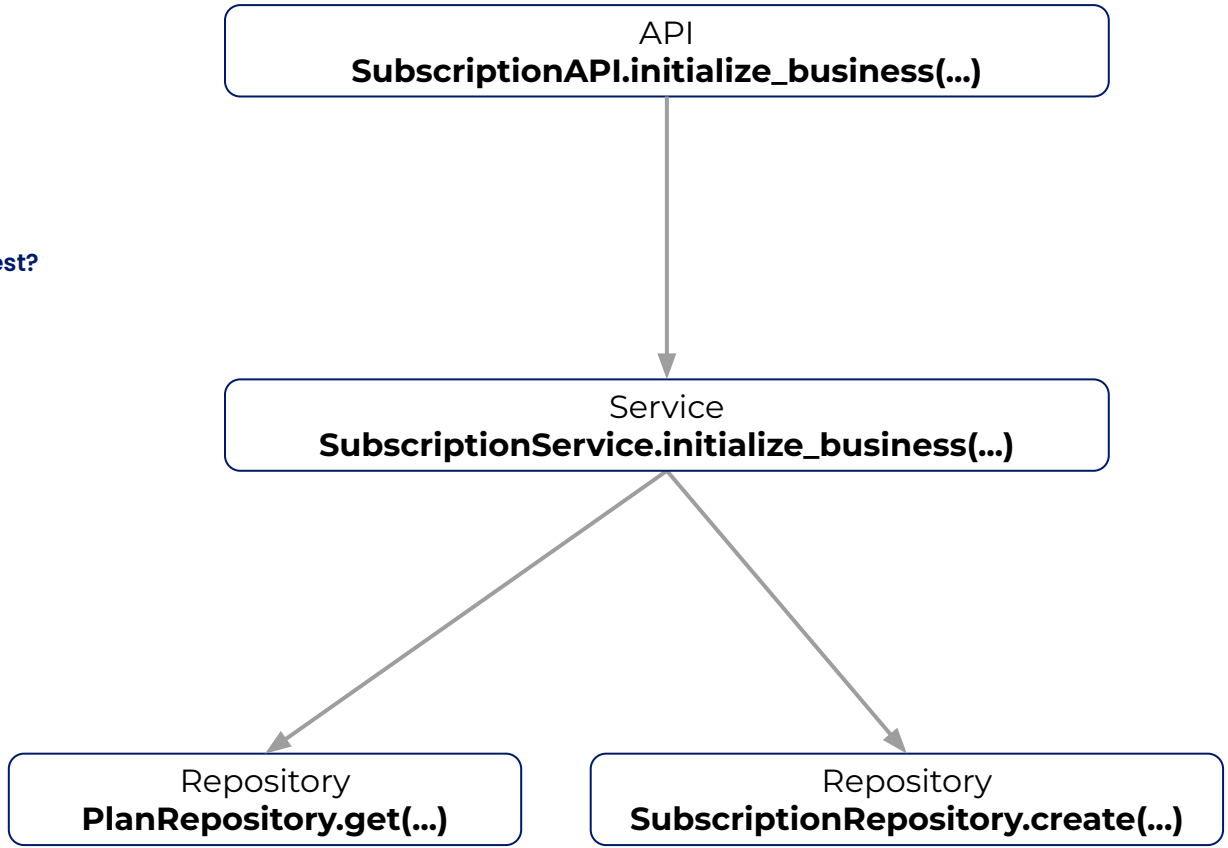
Pilot

ACCELERATE INSTRUMENTATION FOR 1 NEW MICROSERVICE



COMPONENT INTERACTION

What components collaborate to fulfill a request?

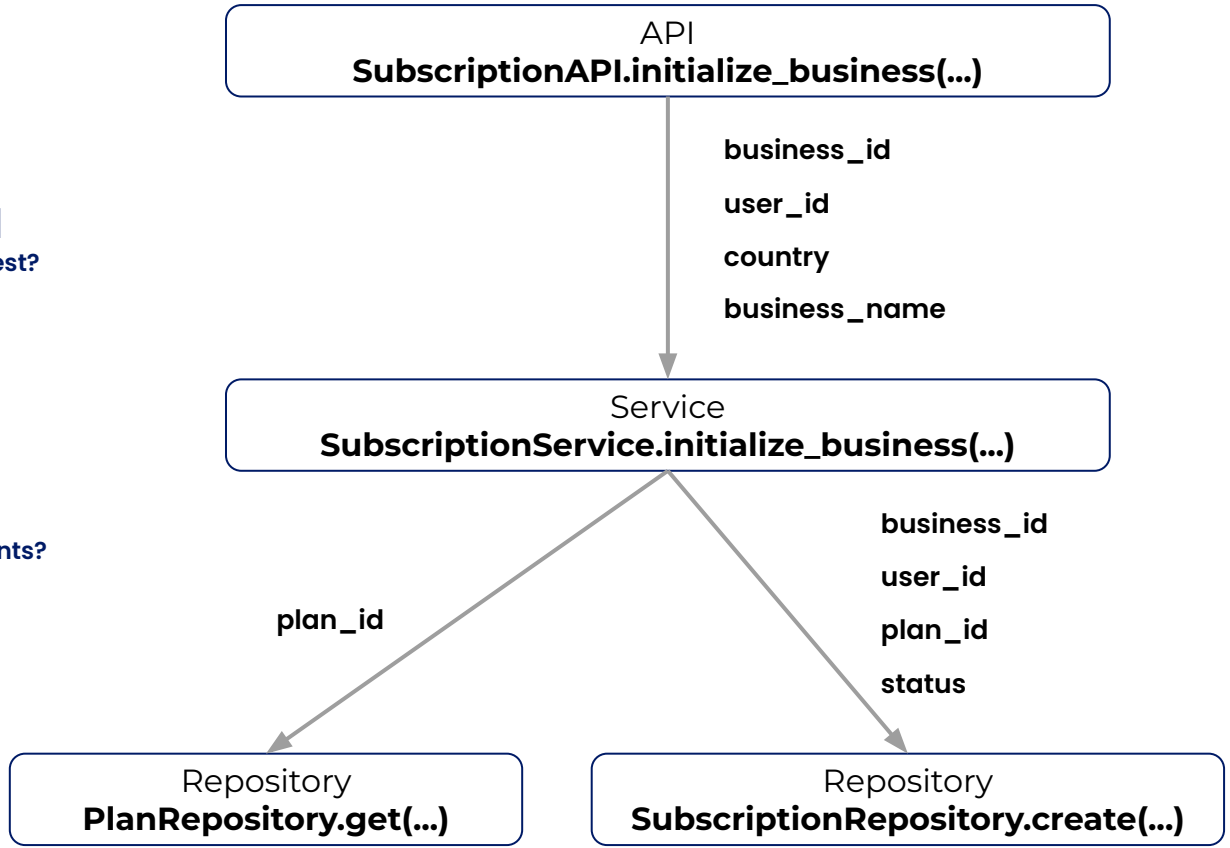


COMPONENT INTERACTION

What components collaborate to fulfill a request?

DATA EXCHANGE

What data is passed between those components?

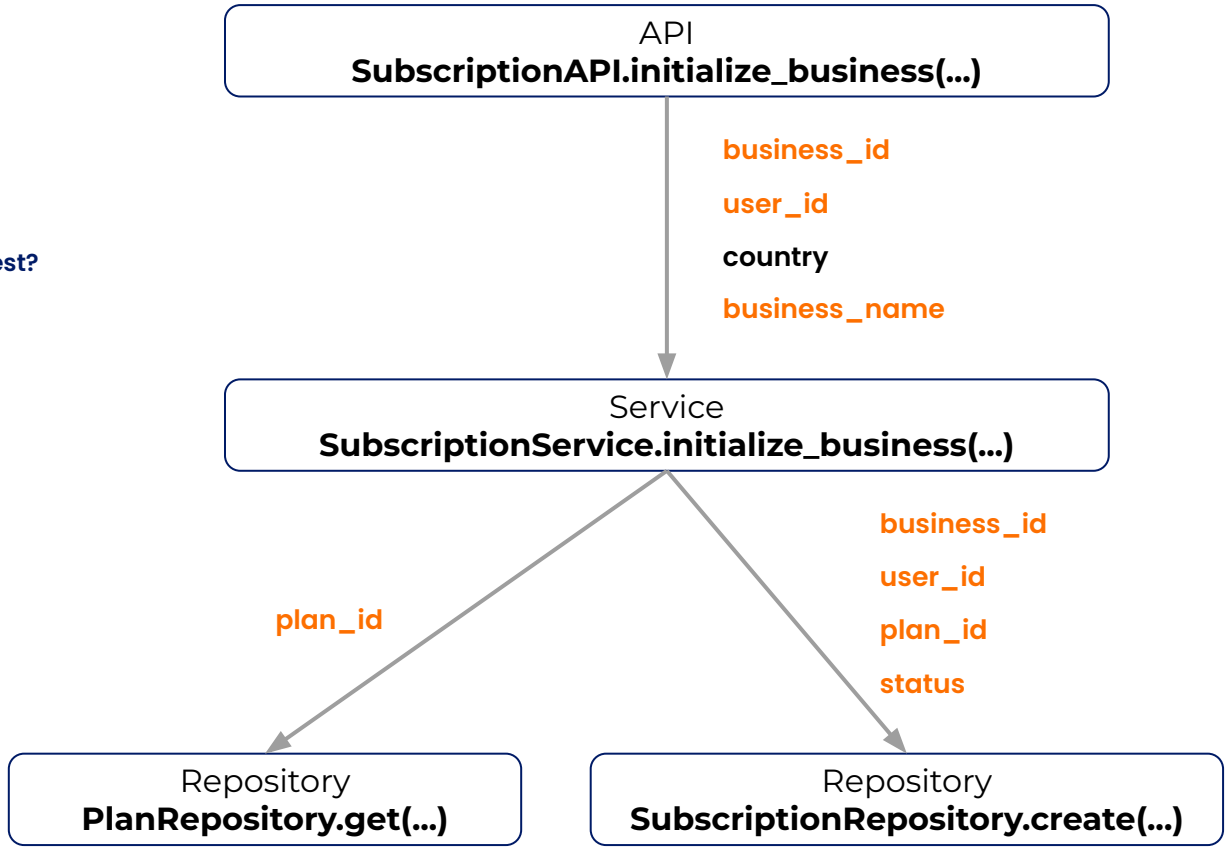


COMPONENT INTERACTION

What components collaborate to fulfill a request?

DATA EXCHANGE

Let's capture data engineers are interested in

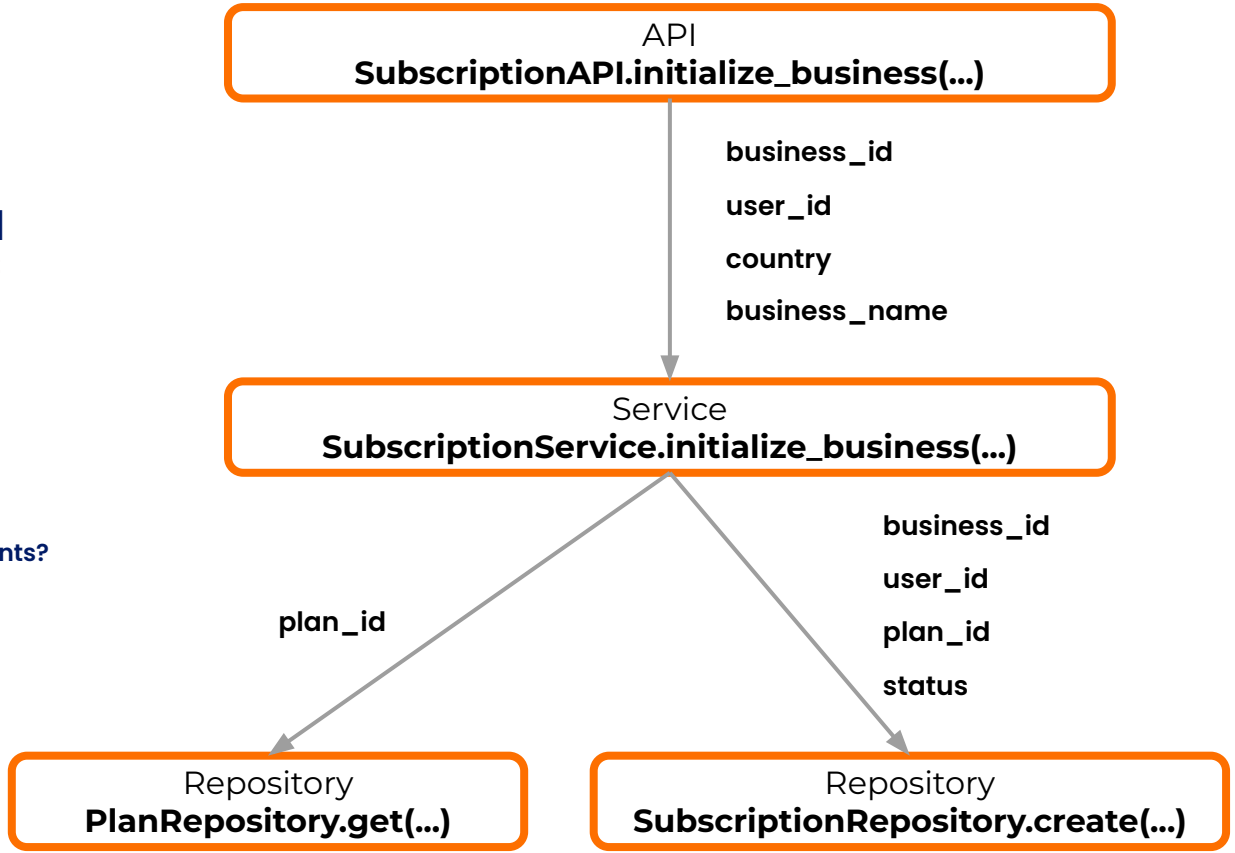


COMPONENT INTERACTION

Let's add some logic to wrap each component

DATA EXCHANGE

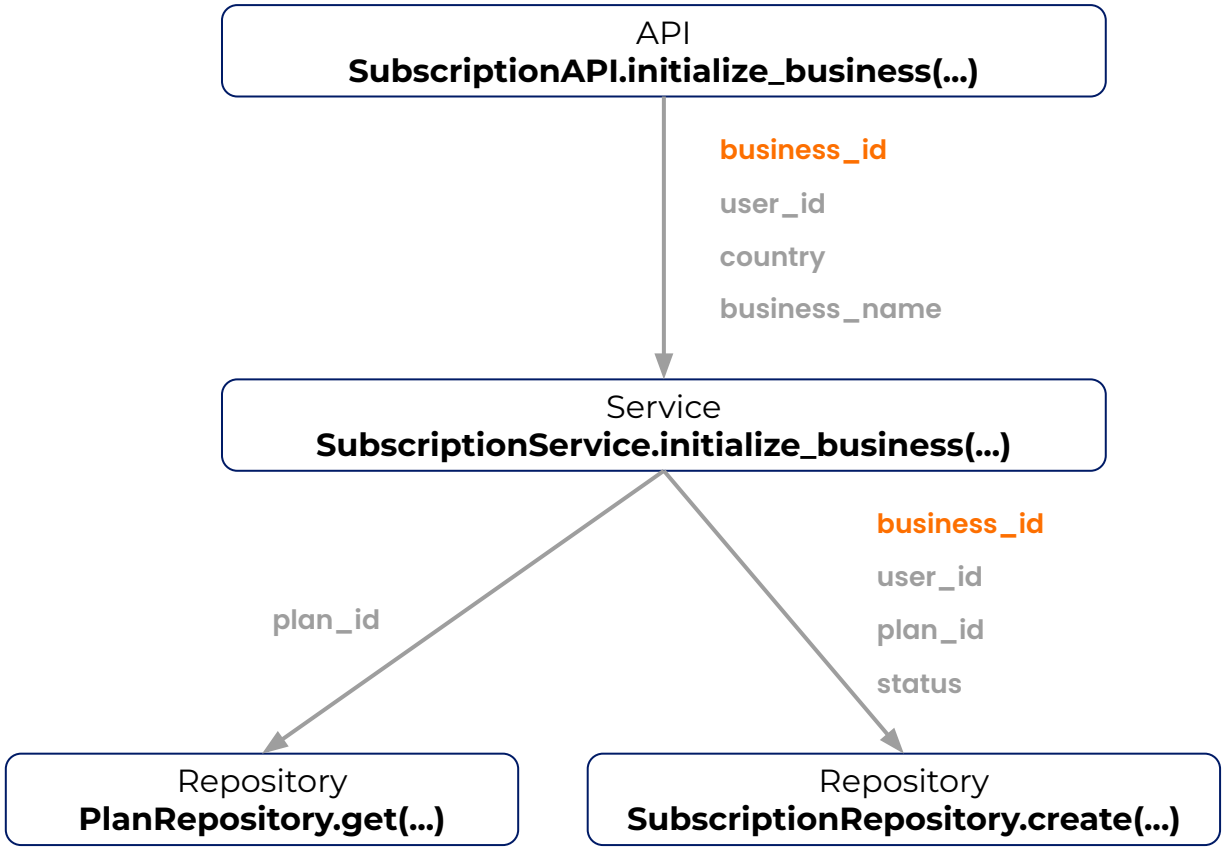
What data is passed between those components?



CONFIGURATION

What data do you want to capture?

business_id

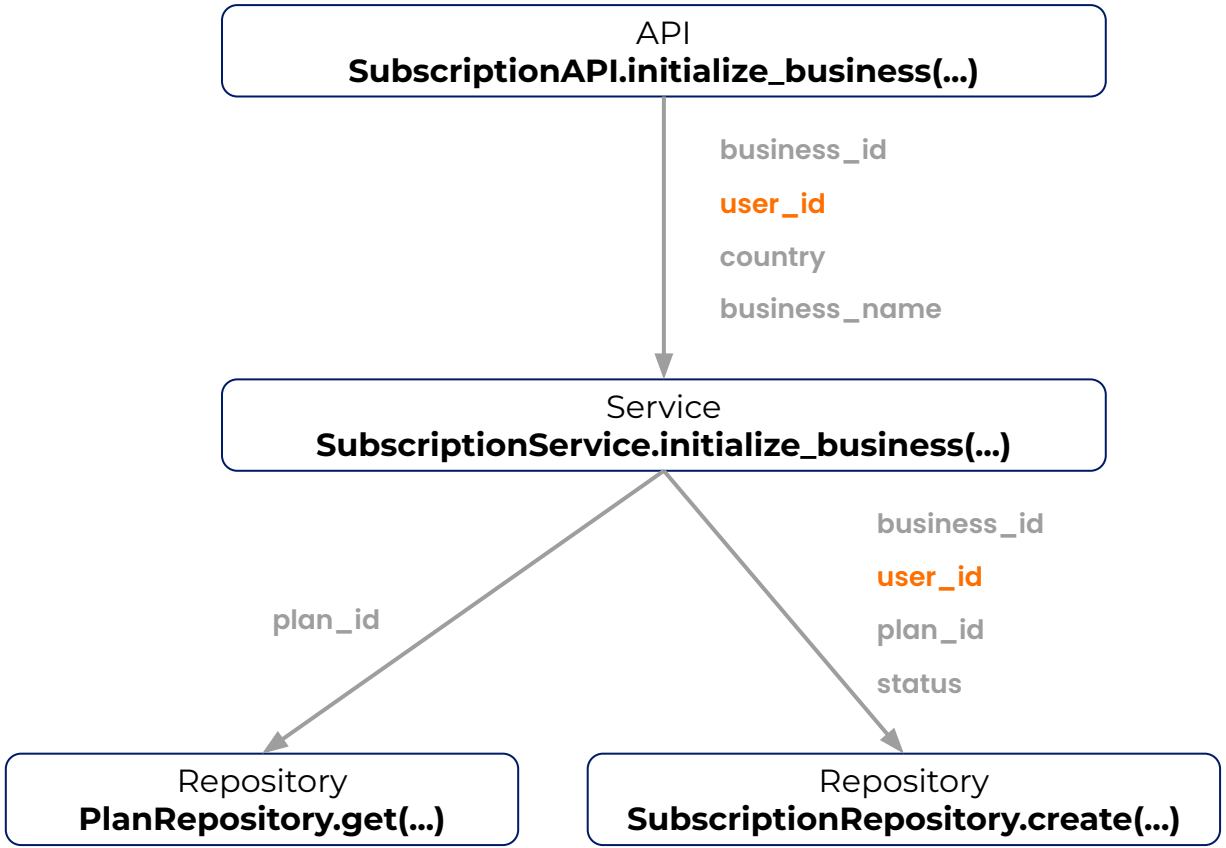


CONFIGURATION

What data do you want to capture?

business_id

user_id



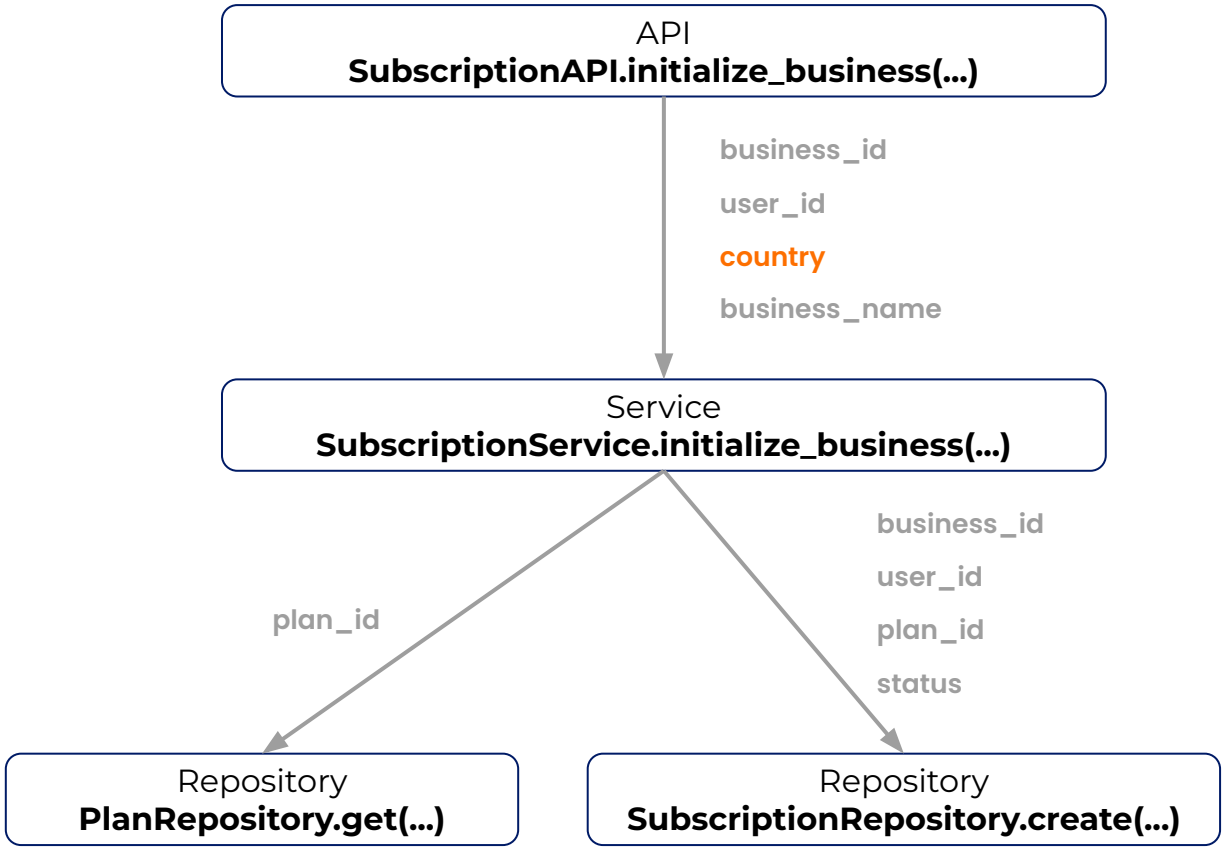
CONFIGURATION

What data do you want to capture?

business_id

user_id

country



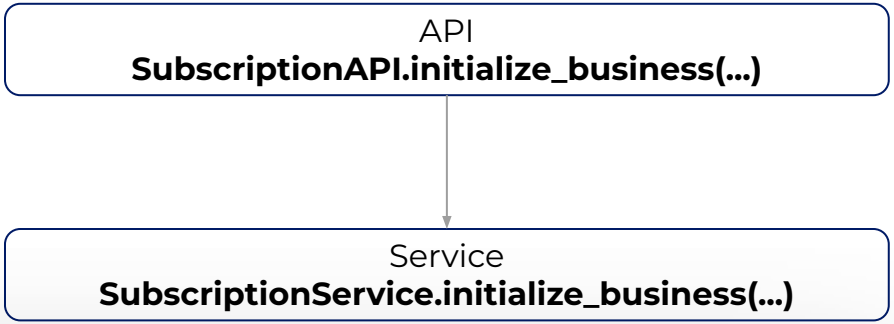
```
TELEMETRY_CAPTURE_CONFIGURATION = TelemetryCaptureConfiguration(
    allowed_field_names=[
        "business_id",
        "user_id",
        "status",
        "plan_id",
        "country",
        ...
    ],
    allowed_field_types=[UUID, datetime, date, bool, Enum],
    allowed_dataclasses=[...],
    allowed_dictionaries=[...],
)
```

SAMPLE CONFIGURATION

Tell us the data you want to capture!

COMPONENT WRAPPING

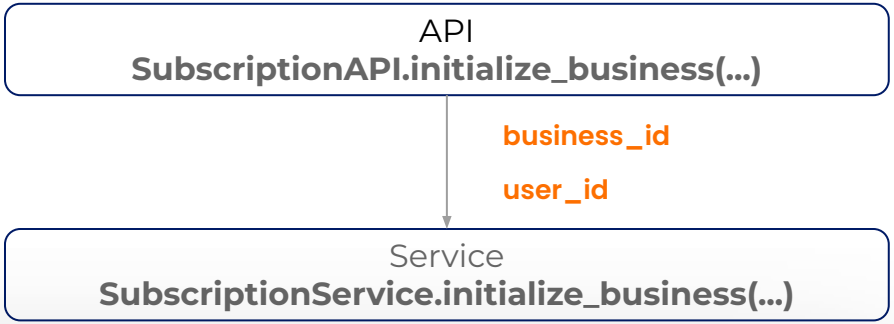
Let's capture the data before calling your method




```python
class SubscriptionAPI:

  def initialize_business(self, business_id, user_id):

    # We need some logic here to capture
    self.subscription_service.initialize_business(business_id, user_id)

    ...
```

COMPONENT WRAPPING

Let's capture the data before calling your method




```python
class SubscriptionAPI:

  def initialize_business(self, business_id, user_id):

    # We need some logic here to capture
    self.subscription_service.initialize_business(business_id, user_id)

    ...
```

COMPONENT WRAPPING

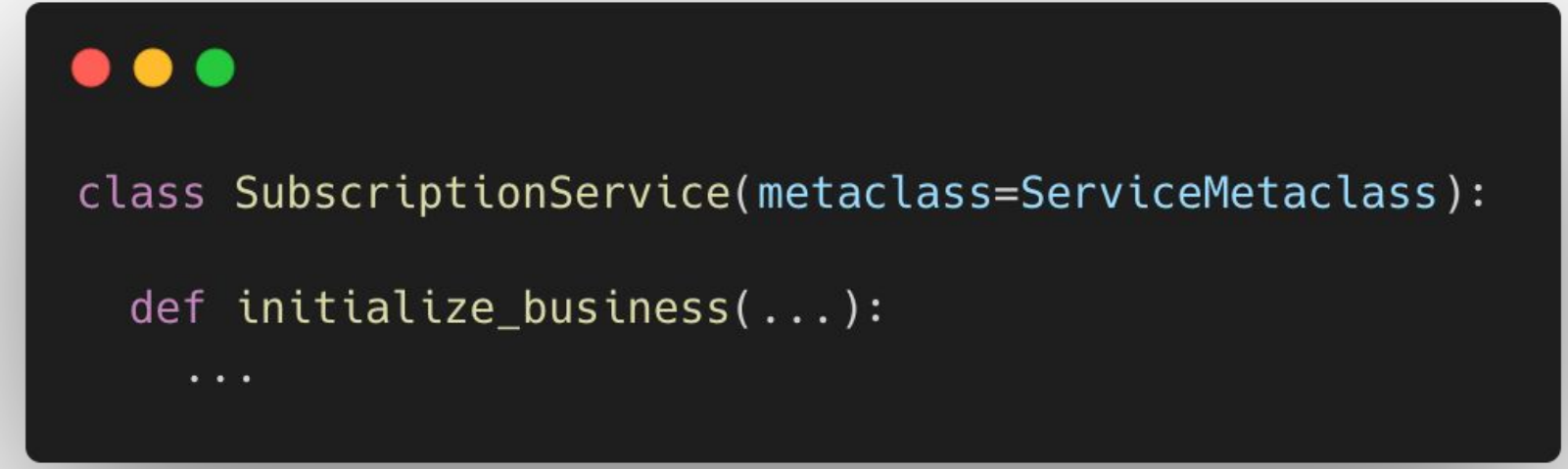
Let's capture the data before calling your method

- Redefine methods
- Capture telemetry
- Call original function

```
class ServiceMetaclass(type):

    def __new__(...):
      for function in functions:

        # Re-define each function

        def new_function_that_captures_telemetry(...):

          # Capture telemetry based on configuration

          # Call the original function
          result = function(*args, **kwargs)


      # Overwrite the original function implementation
      functions[function_name] = new_function_that_captures_telemetry

      return type.__new__(...)
```

COMPONENT WRAPPING

Use the metaclass



COVERAGE REPORT

How do I know what'll be captured?

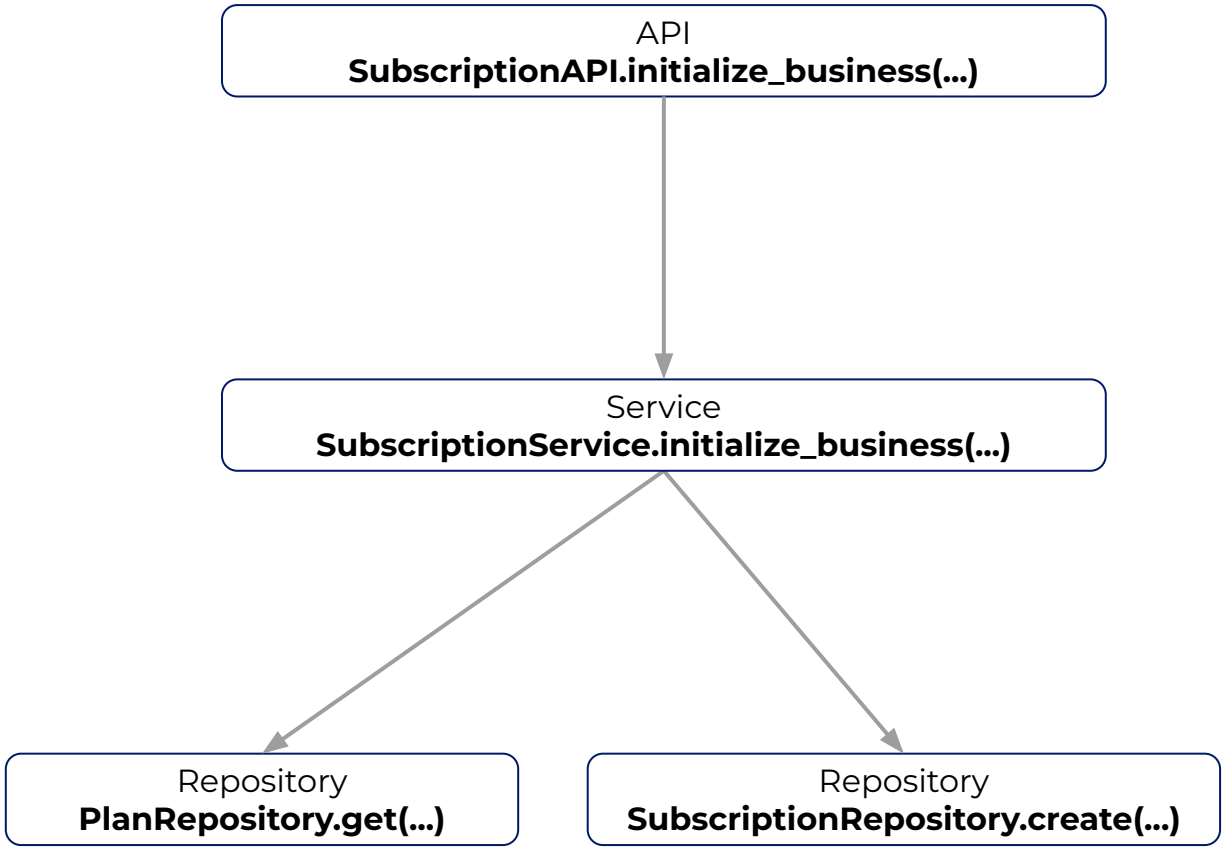
Printing Observability Coverage Report 4s

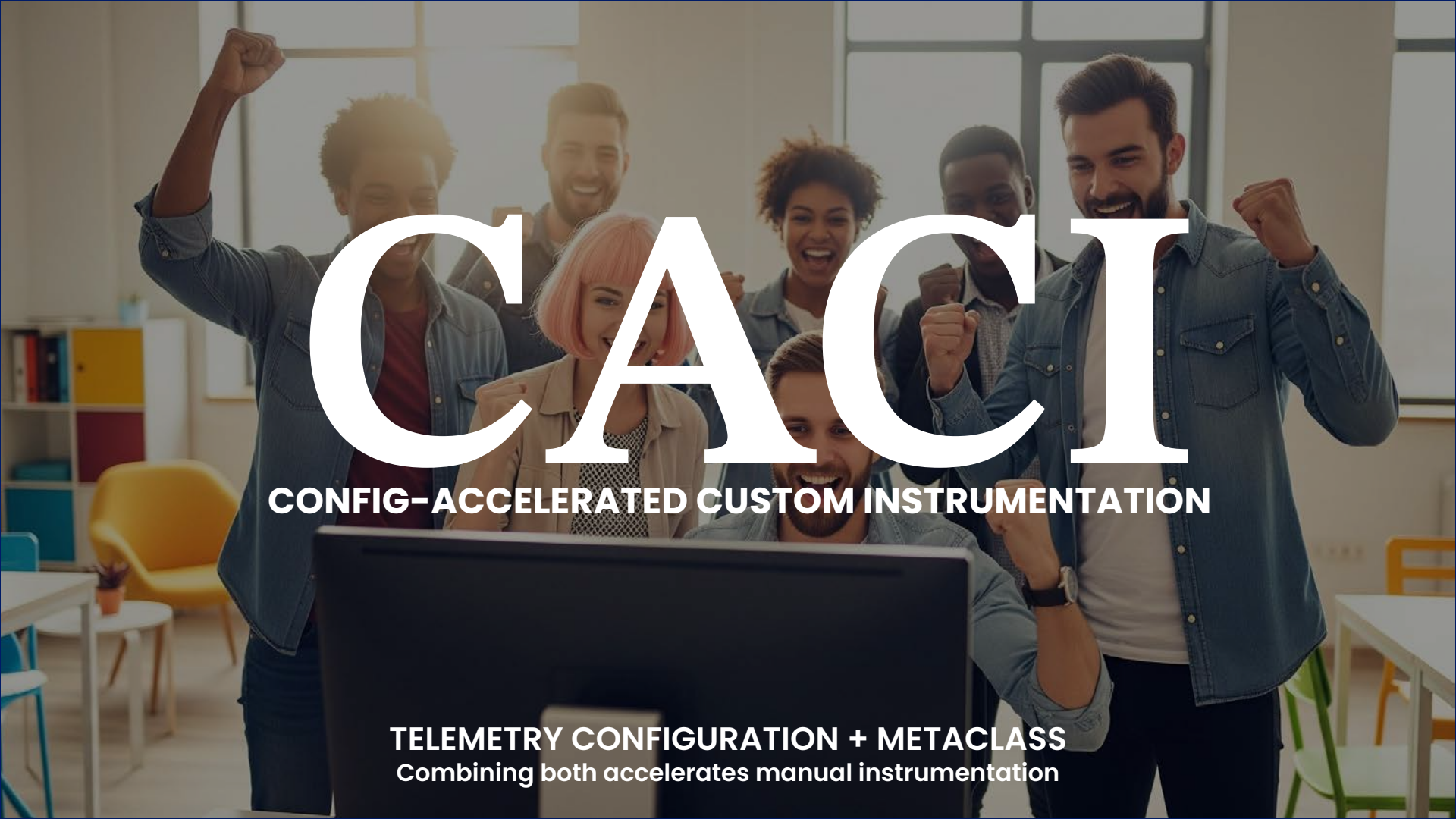
```
#!/bin/bash -eo pipefail
./manage.py observability_coverage_report || true
```

```
- BankAccountService
  - add_error(self, field: str, message: str) -> None
  - Inputs
    - field:str - 🚫
    - message:str - 🚫

  - all_errors(self) -> dict[typing.Any, typing.Any]
  - Inputs

  - create_bank_account_check(self, ctx: dict[str, typing.Any], business_id: payroll.datatypes.business.BusinessId, bank_account_id: Optional[payroll.datatypes.bank_account.BankAccountId] = None, employer_id: Optional[payroll.datatypes.employer.EmployerId] = None, employee_id: Optional[payroll.datatypes.employee.EmployeeId] = None, contractor_id: Optional[payroll.datatypes.contractor.ContractorId] = None, **bank_account_attrs: Any) -> Optional[payroll.datatypes.bank_account.EmbeddedBankAccount]
  - Inputs
    - business_id:UUID - ✅
    - bank_account_id:int - ✅
    - employer_id:int - ✅
    - employee_id:int - ✅
    - contractor_id:int - ✅
    - bank_account_attrs:Any - 🚫

  - get_bank_account_check(self, ctx: dict[str, typing.Any], check_bank_account_id: str) -> Optional[payroll.datatypes.bank_account.EmbeddedBankAccount]
```

TELEMETRY CAPTURE

We'll capture the data from any component!

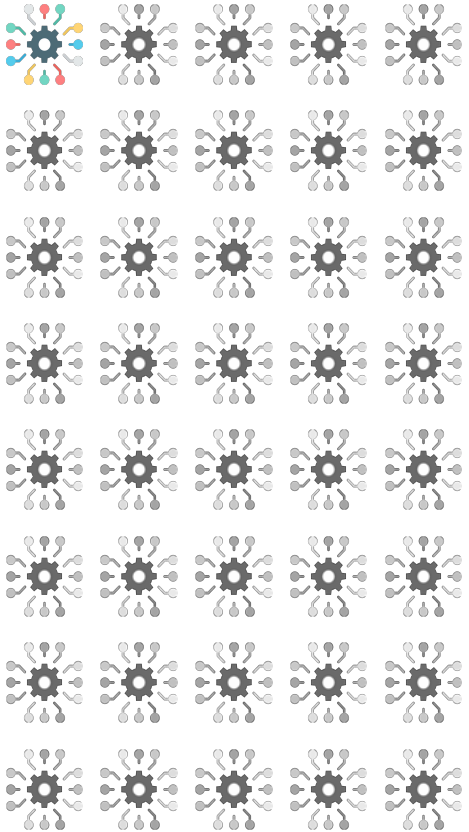



CACI
CONFIG-ACCELERATED CUSTOM INSTRUMENTATION
TELEMETRY CONFIGURATION + METACLASS
Combining both accelerates manual instrumentation

1% → 60%

INSTRUMENTATION

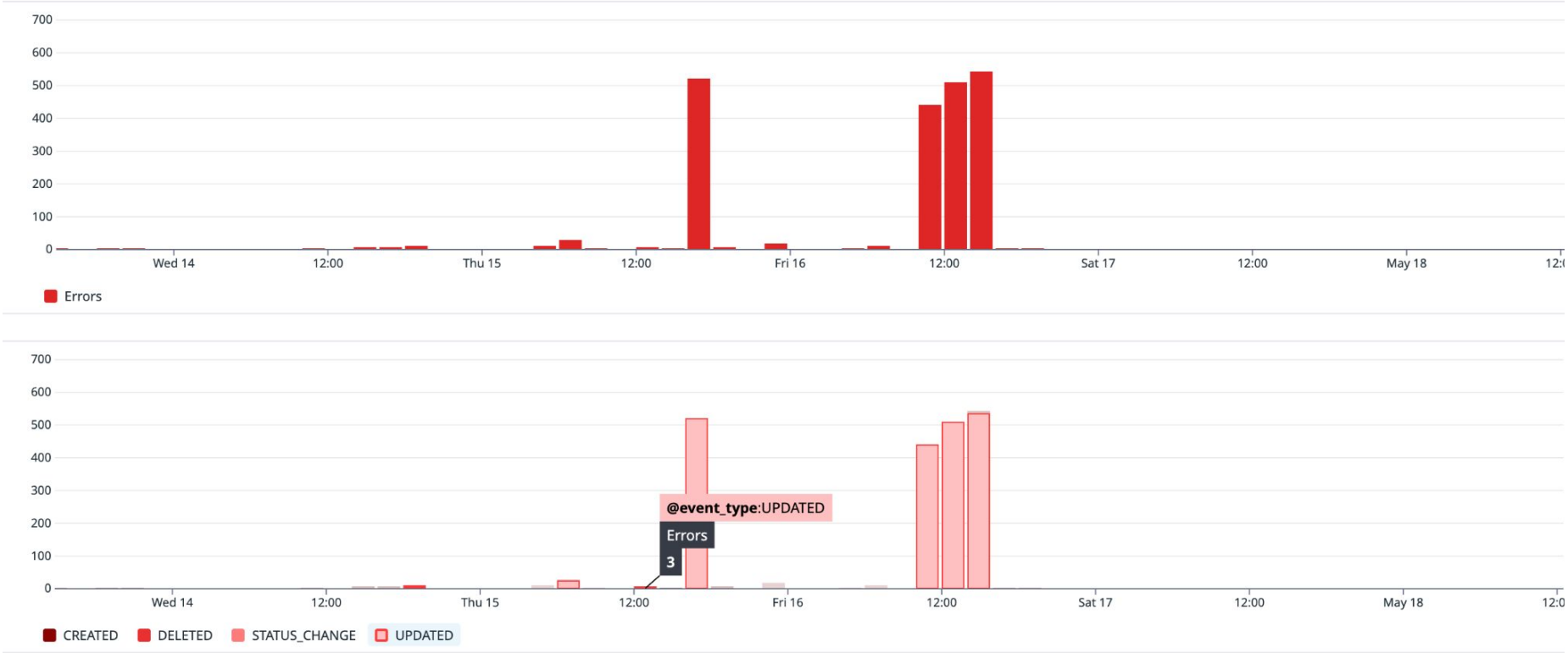
INSTRUMENTATION
~1 week



Pilot

ACCELERATE INSTRUMENTATION FOR 1 NEW MICROSERVICE

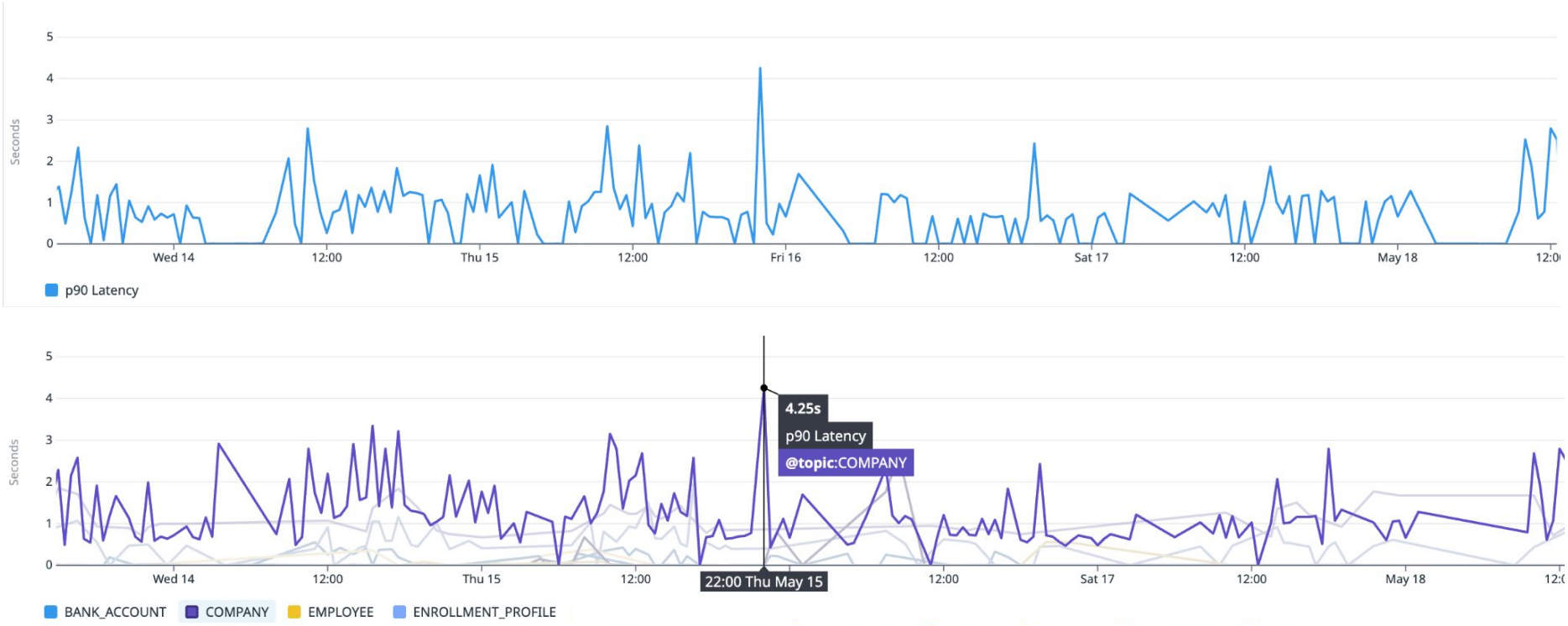
Event Processing Errors



ERRORS?

Let’s be more specific actually

Event Processing Latency

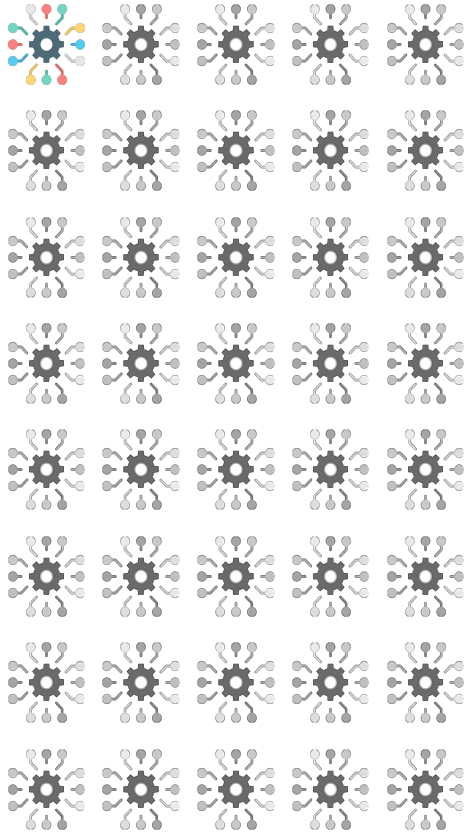


LATENCY?

Let’s be more specific actually

What's next?

- Scaling up for the next 39+
- Component agnostic metaclasses
- Output capture
- Using LLMs to accelerate integration
- Exploring similar capabilities in Datadog



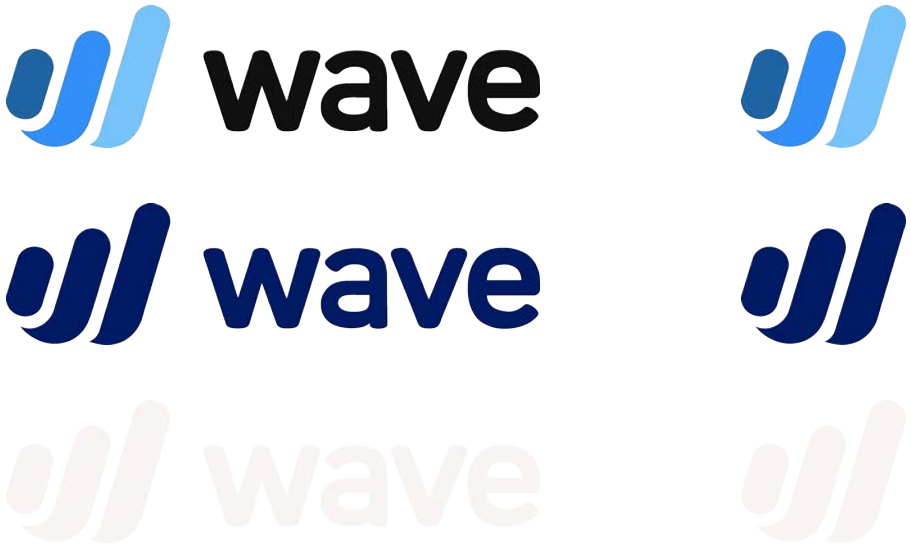
Thank you

Jean-Mark Wright
Staff Engineer, Observability
Wave Financial Inc

https://www.linkedin.com/in/jean-mark-wright/

https://jaywhy13.hashnode.dev/

Logo & Icon



Flourishes

Colours


Midnight Ledger
001b66
Brilliant Balance
0040ff
Invoice White
f9f4f1
Soft Dividend
e8aaff
Fiscal Sky
328ff8
Financial Breeze
76c3fc
Wealth Violet
8f4ae2
Golden Return
fdc237
Revenue Flame
fd6f00
Growth Green
008238
Capital Moss
0c4527