MODERNIZING LEGACY CODE - WHILE NOT BRINGING DOWN PROD

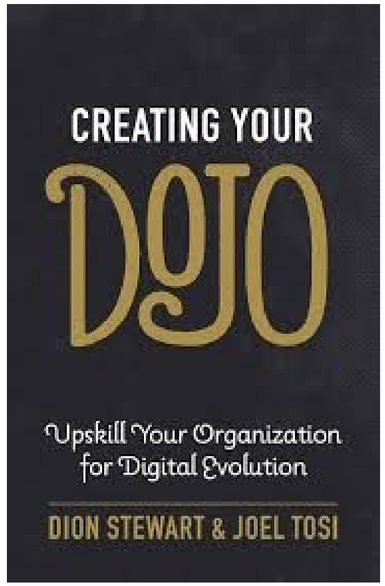
Joel Tosi

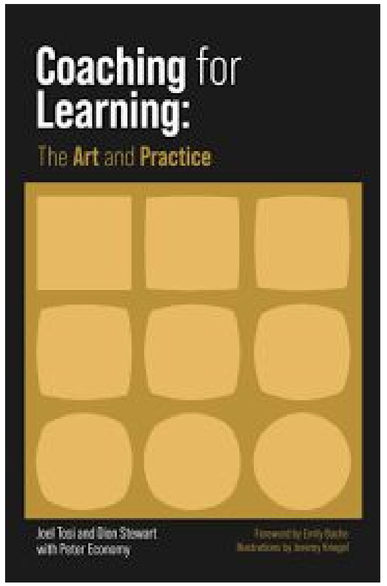
Slides – https://www.dojoandco.com/speaking

@joeltosi@mastodon.social

Joel.Tosi@dojoandco.com

ABOUT ME





CME Group

verizon

ATLASSIAN

CENTENE Corporation

Northwestern Mutual

IT

Coming Fall 2026

MARK43

Edward Jones

TVA

STELLANTIS

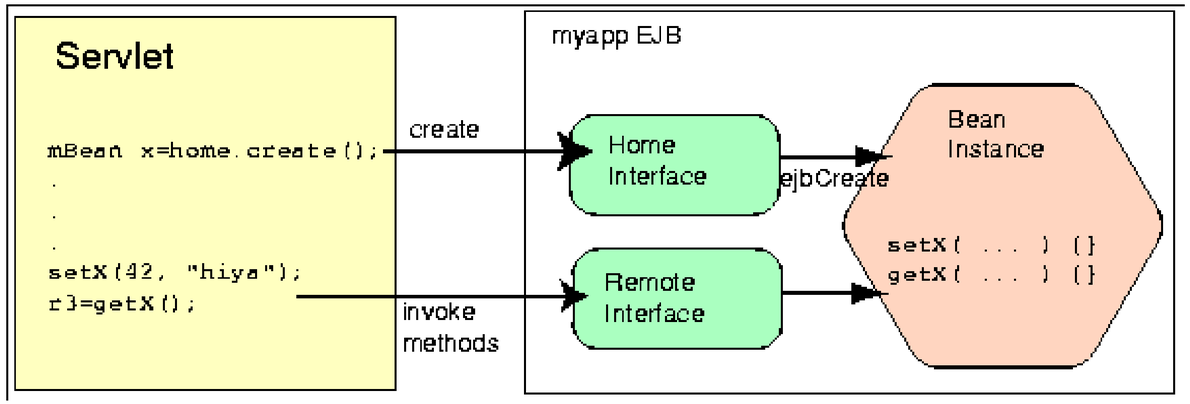
FCA
FIAT CHRYSLER AUTOMOBILES

Human Resources and Social Development Canada
Ressources humaines et Développement social Canada

NG

TO ANSWER ‘WHAT IS MODERNIZING’ - WE MUST KNOW WHY ARE WE MODERNIZING?

WHY MODERNIZE?

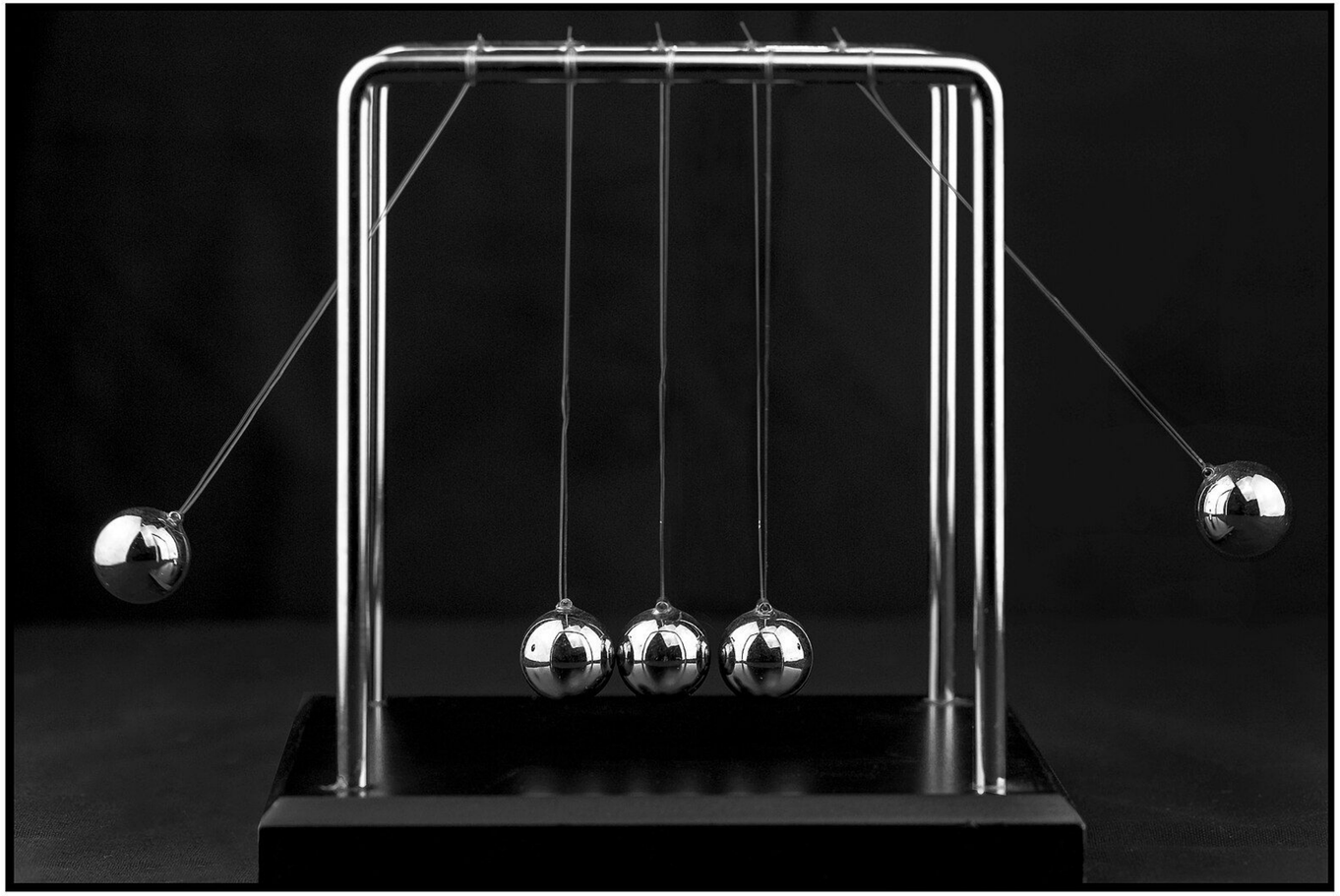


OTHER REASONS TO MODERNIZE

- Product and Architecture are no longer aligned
- Product and Code are no longer aligned
- Product and Organization are no longer aligned

RESULTING IN

- High Costs (Coordination, Defects, etc)
- Lower Innovation
- Slower Learning

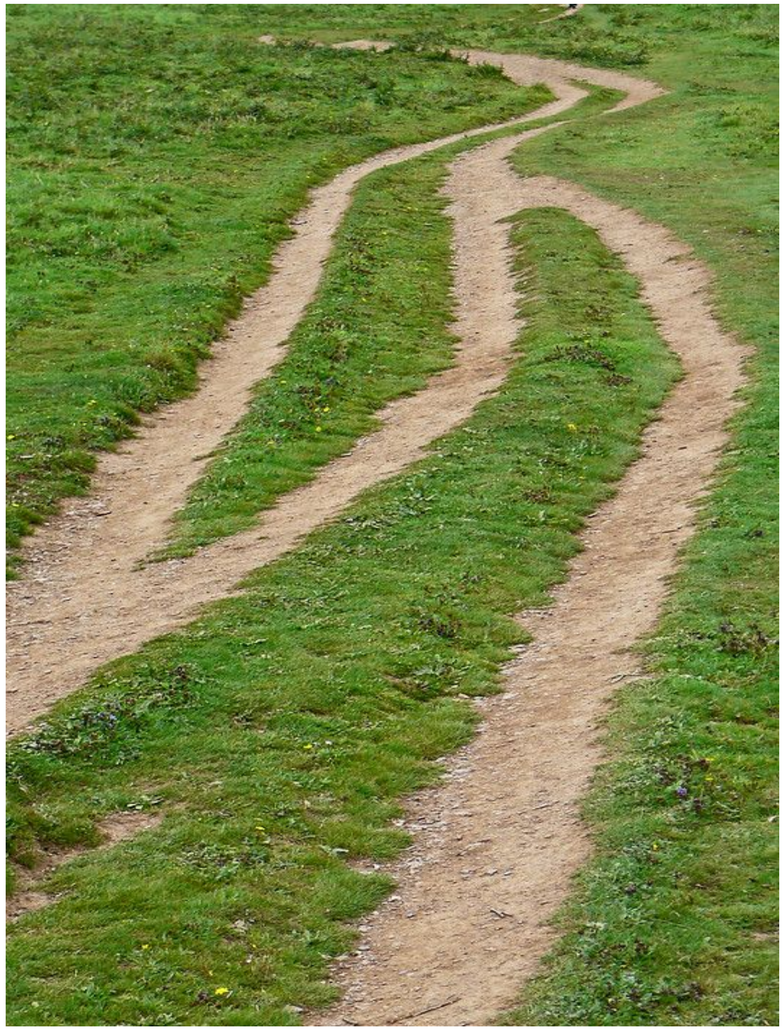


Sheila Sund from Salem, United States, CC BY 2.0 <https://creativecommons.org/licenses/by/2.0>, via Wikimedia Commons

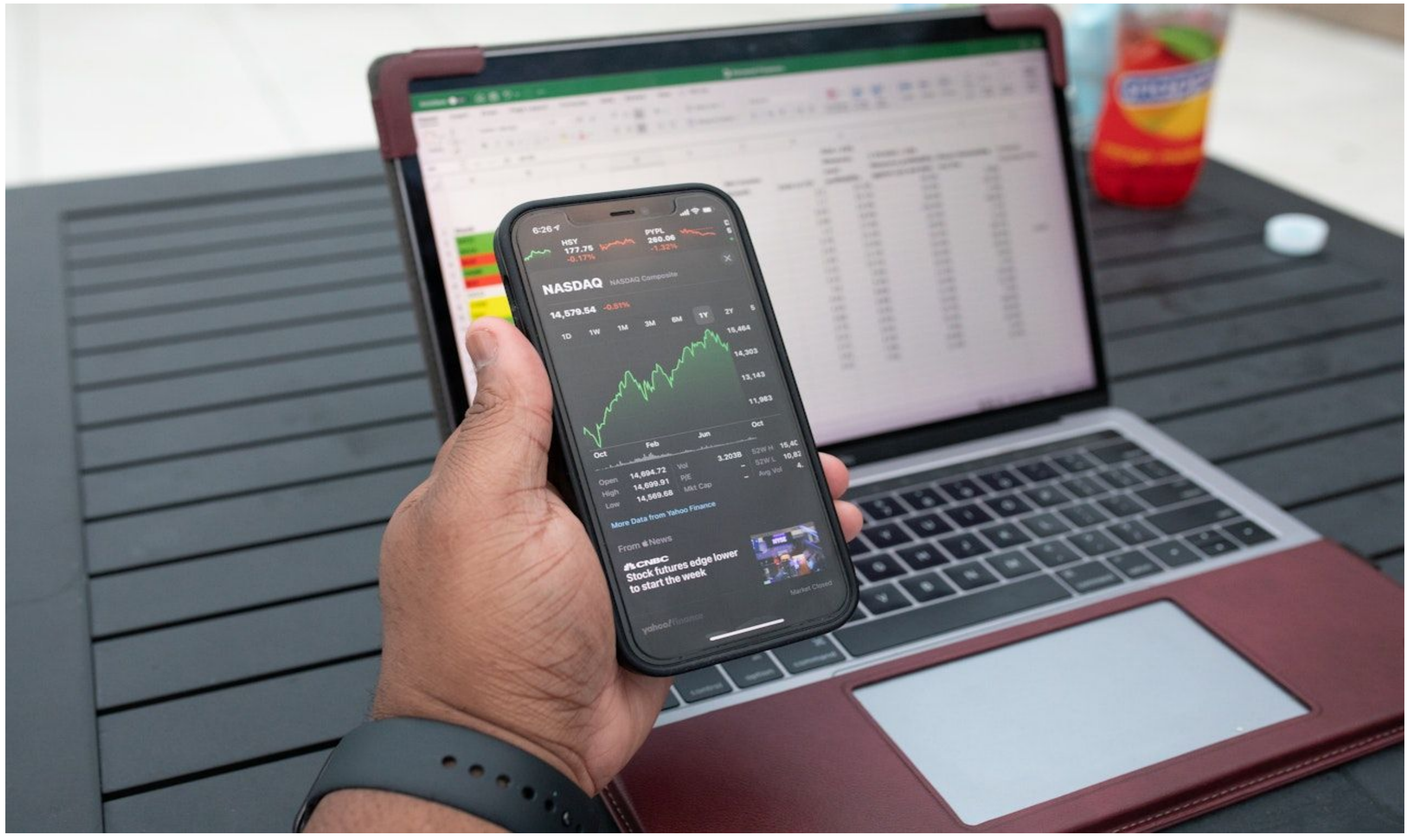
MODERNIZATION IS ENABLING BETTER PRODUCT EXPERIENCES AND EFFECTIVE PRODUCT DELIVERY THROUGH A RETHINK OF TECH

WHERE WE ARE GOING

- Realigning Product And Arch
- Finding Seams
- Questioning Product Assumptions
- Making Boring Changes
- Expansion / Contraction
- Modernize or Migrate?



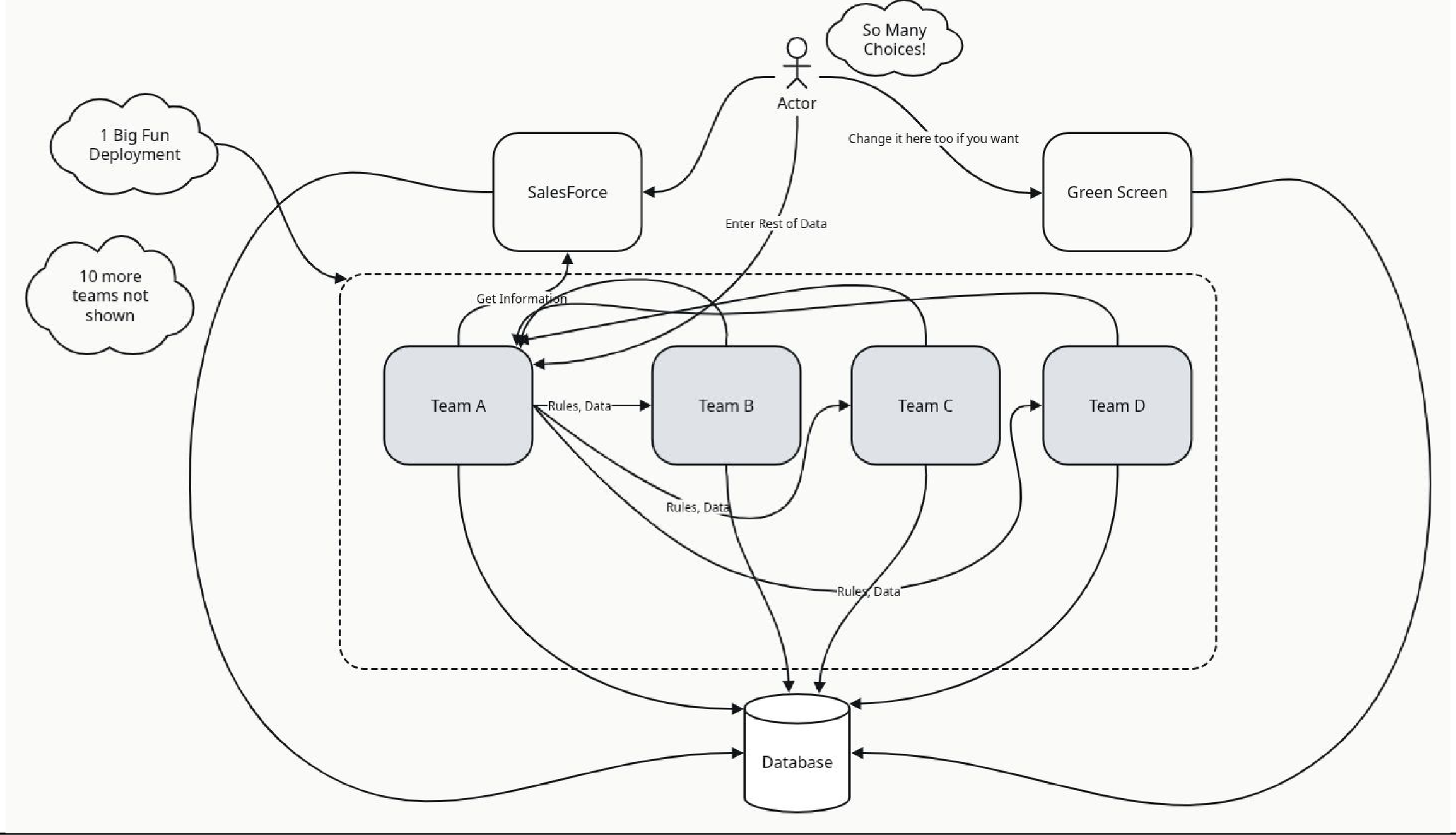
BACKGROUND STORY



https://nappy.co/photo/OHzay15gpm5tHzJD6Hu6A

- 20 years
- 1 Team, 2 teams...14 teams
- Over 1.4M lines of code
- Struts / JSP
- EJBs!
- WebLogic
- DB2
- Mainframe

REALIGNING PRODUCT AND ARCHITECTURE



DIFFICULT ... TO DEBUG, SHIP, INNOVATE

Did not happen overnight

Way more than technical debt

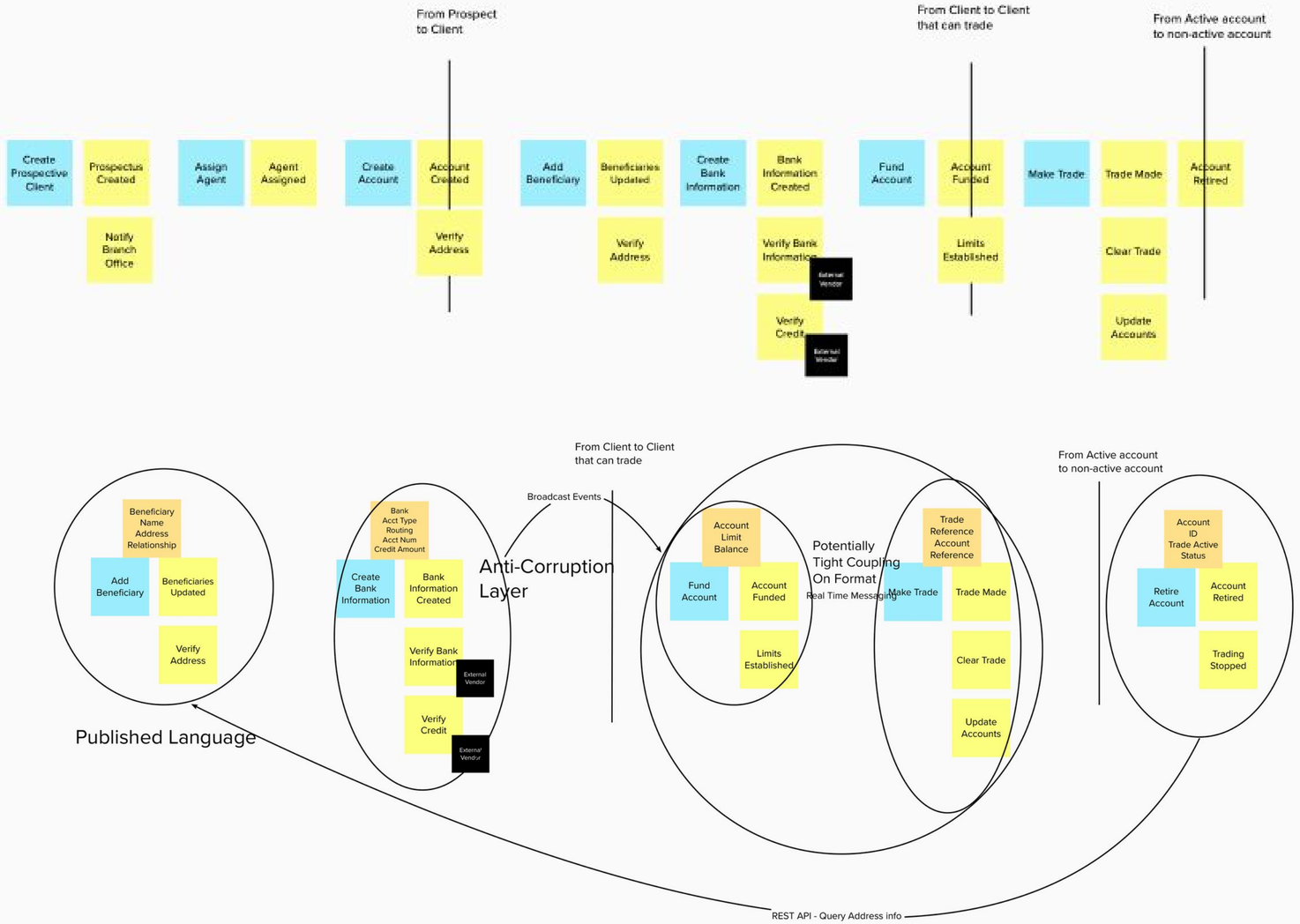
Org structure

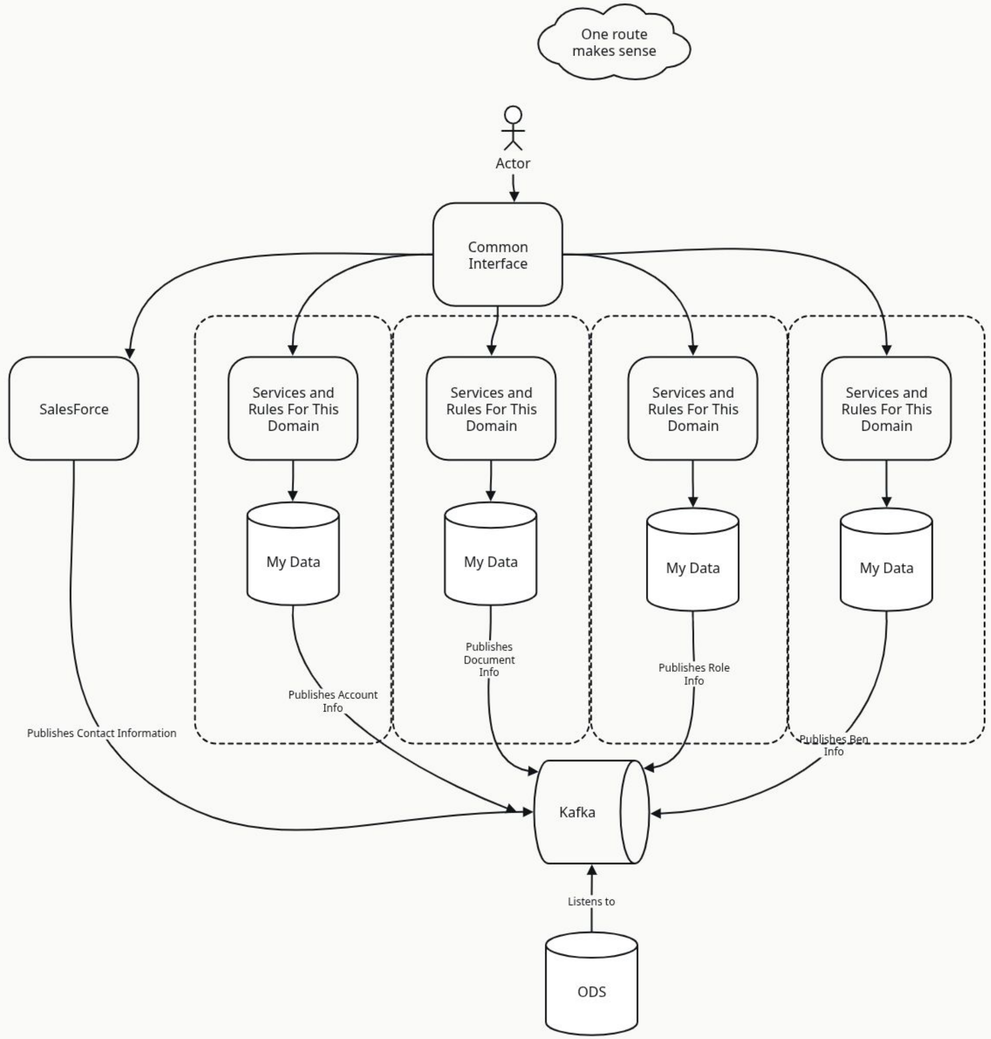
Process

Environments

Lady standing in the scorching sun is crying in silence, she may be waiting for someone..Something bad happens to her ....everything is so #blackandwhite to her @ 人民广场 | People's Square 4sq.com/17kptqh (posted via FlickSquare)

STRATEGY TO ALIGN PRODUCT / ARCH / ORG





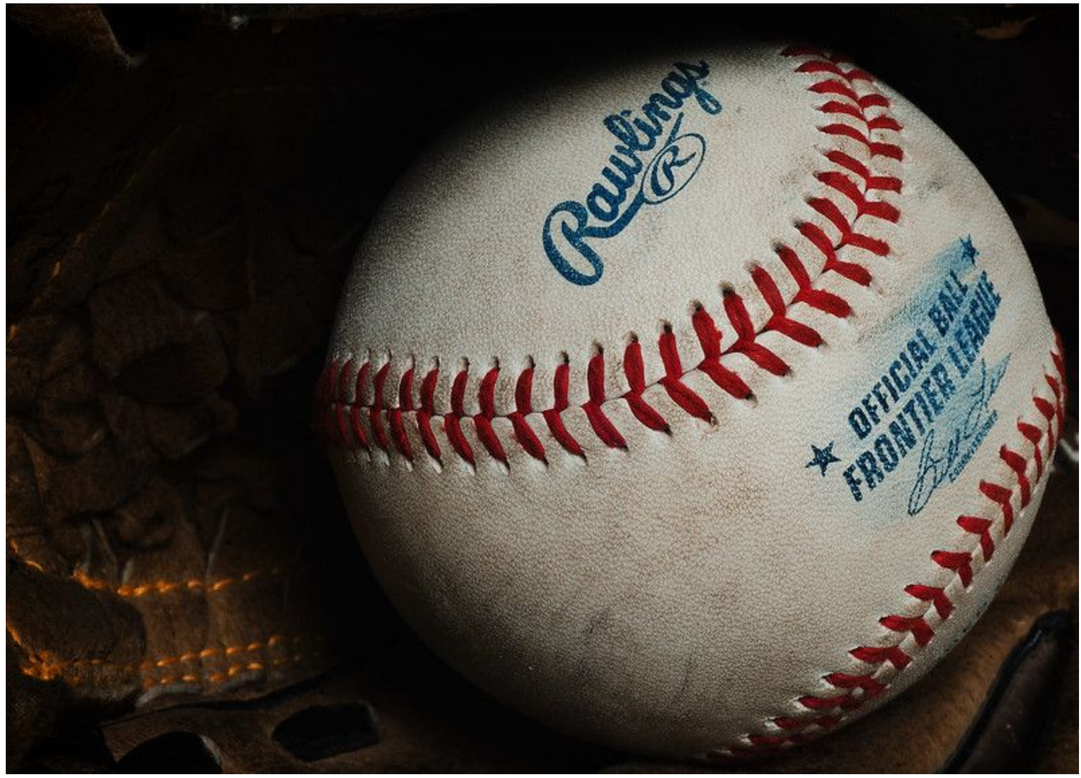
FINDING SEAMS IN THE PRODUCT

FIND SEAMS

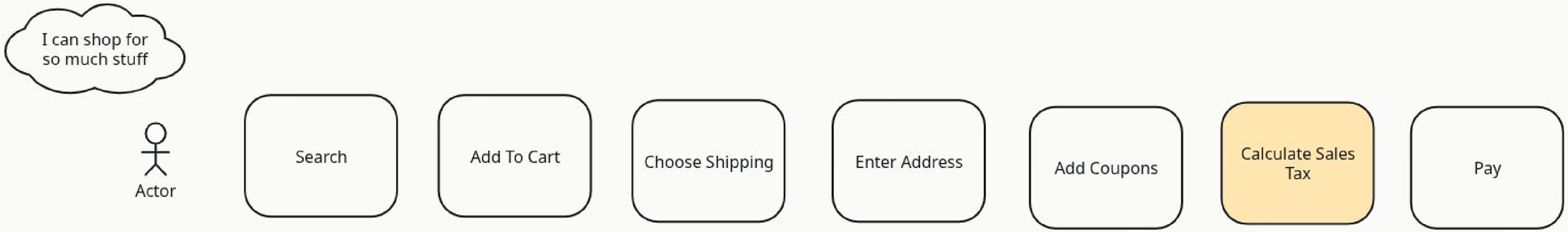
A Seam in a product is a point where you can divert to a new experience

A Seam in code is an inflection point where you can divert to a new behavior or tech without editing that spot

With both seams, your goal is to minimize impact



SEAMS IN PRODUCT (SEGMENT)



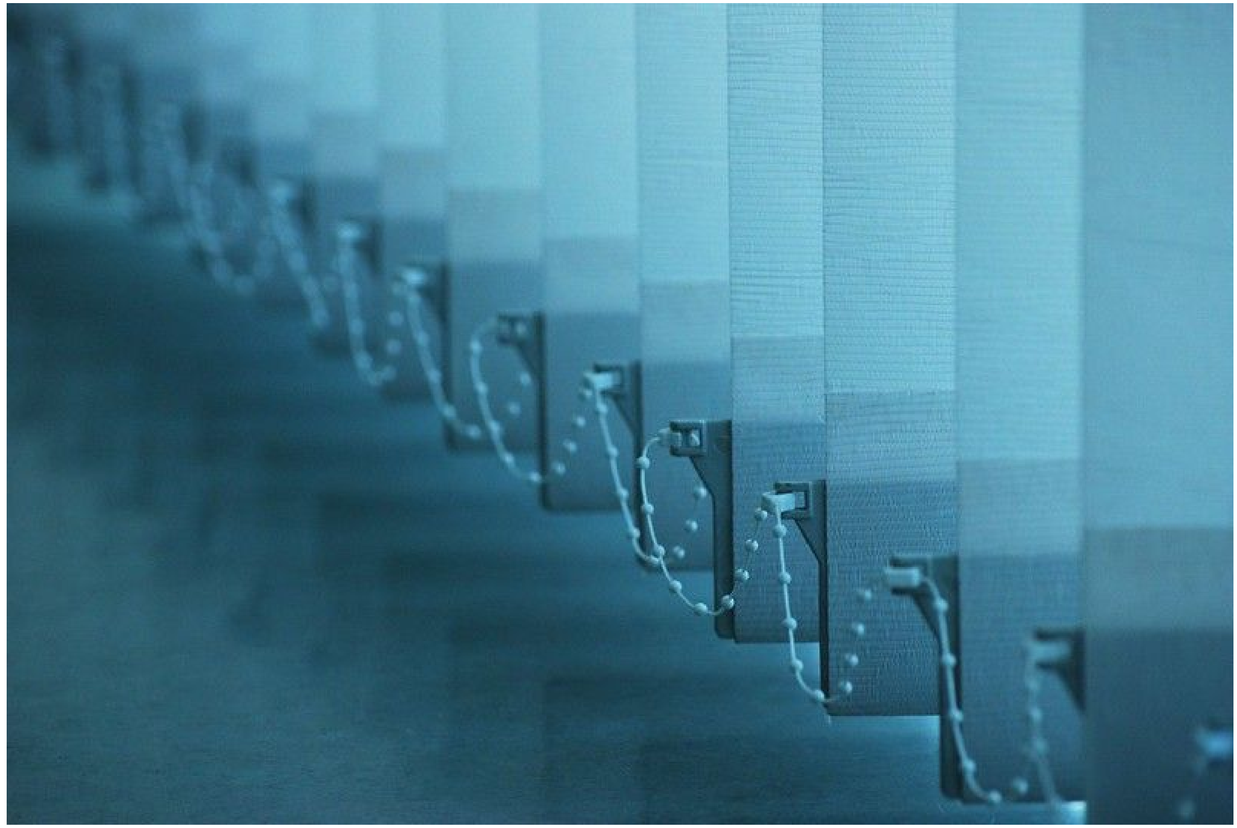
SEAMS IN TECH

SEAMS AREN'T ALWAYS AN OPTION

Seams are great - when they exist (usually do)

Sometimes you need another strategy

Parallel and Compare



QUESTIONING PRODUCT ASSUMPTIONS (AND CODE)

LEGACY PRODUCTS HAVE YEARS OF IDEAS

First - remove dead / inaccessible code

Second - remove code that is never executed

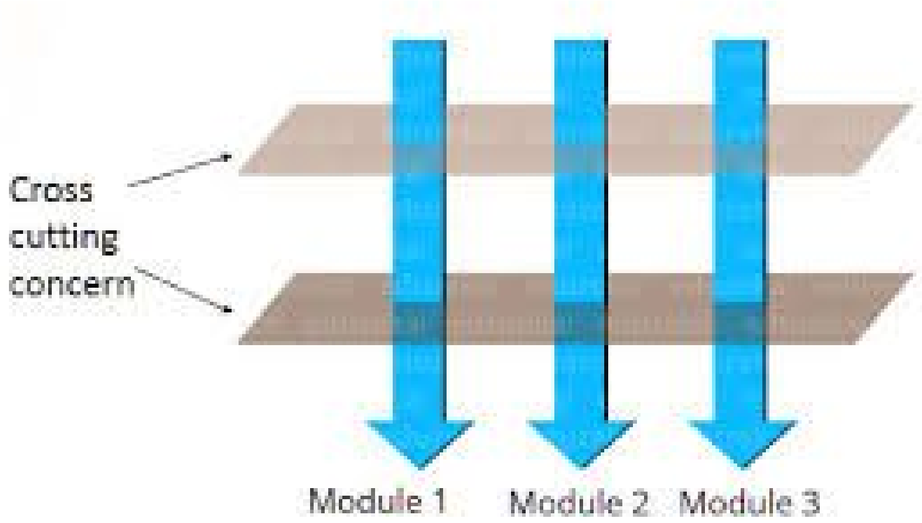
Third - Look for better way (UX)

```
if uid:
    return __execute_cmd('config -g cfgUserAdmin -o \
                          cfgUserAdminUserName -i {0} ""'.format(uid))

else:
    log.warning('\'{0}\' does not exist'.format(username))
    return False

return True
```

Delete this unreachable code or refactor the code to make it reachable. See Rule — 7 days ago

Bug · Major · Confirmed · Not assigned · 5min effort · Comment — cwe, u

OpenTelemetry



DELETING CODE CAN BE SCARY

Work small

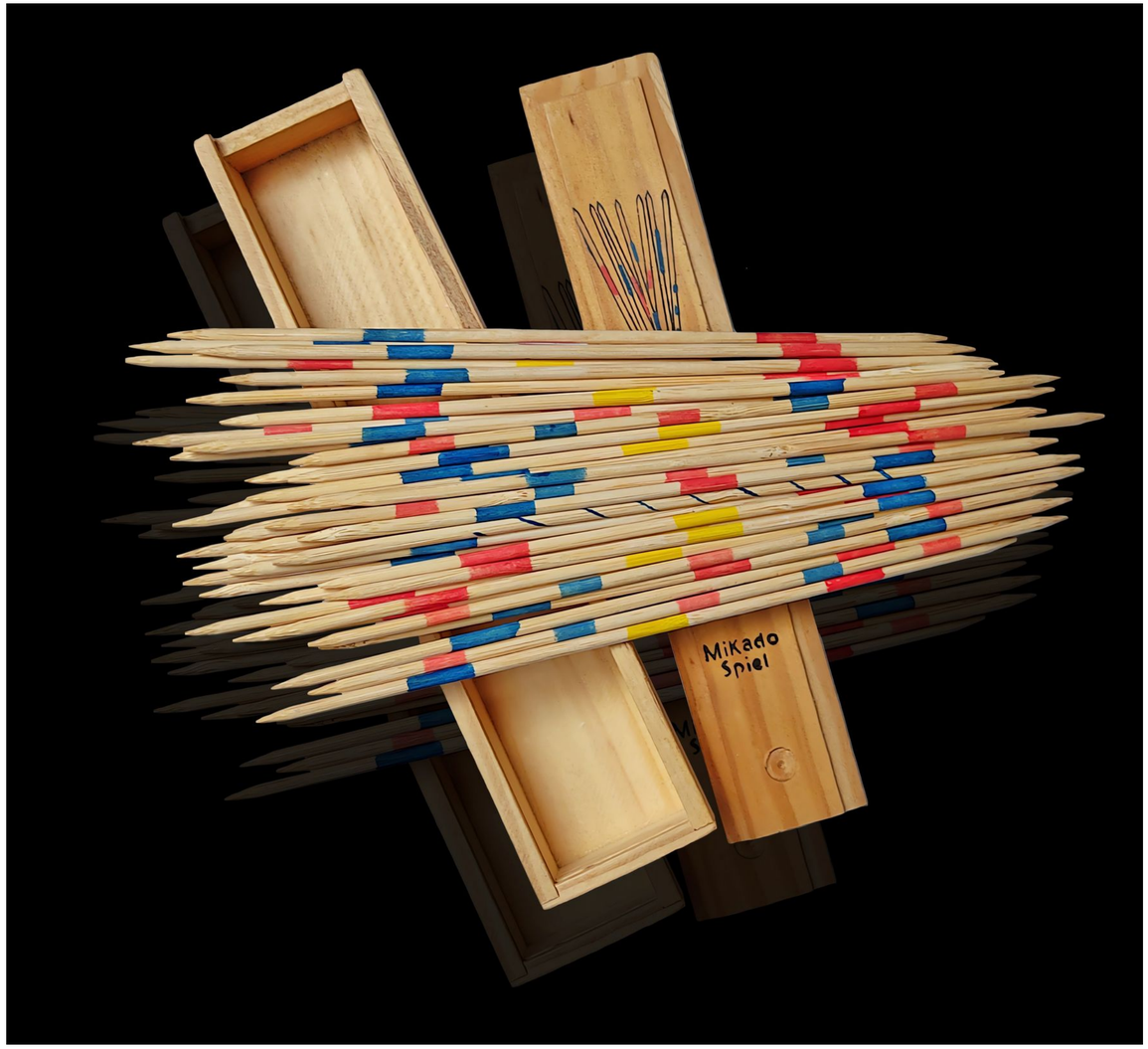
Characterization Testing

Mikado method

Remember you can revert

The easiest code to modernize?

The non-existent code

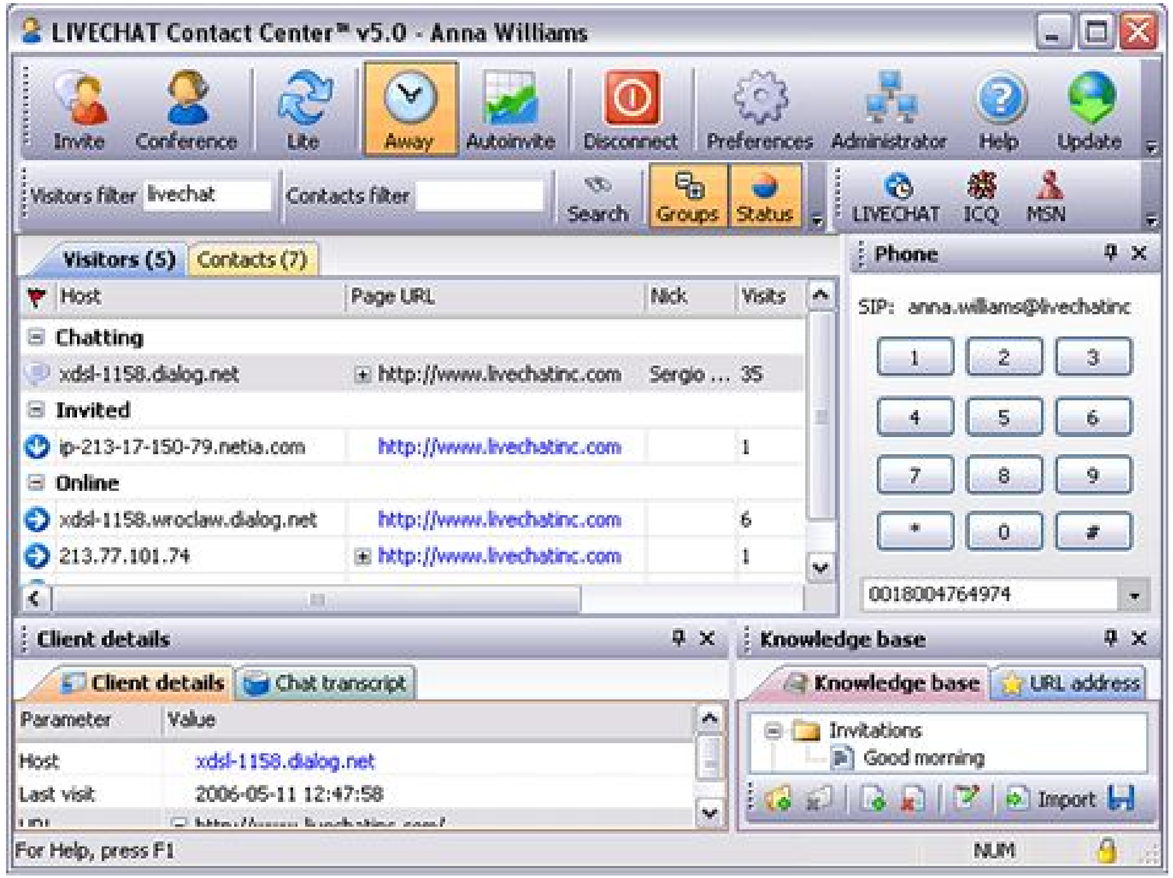


User:Fornax, CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0>, via Wikimedia Commons

PRODUCTS NEED TO BE REFACTORED

Legacy products have years of ideas

- Is that idea still relevant?
- Do people actually use this?
- Is it helpful?
- Is the data actually used?
- Is Feature X / App Y because of a failure somewhere else (General Edit screens)



MAKING BORING CHANGES

BUT IT IS SOOOO BORING

Small, shippable, testable

Didn’t change the database

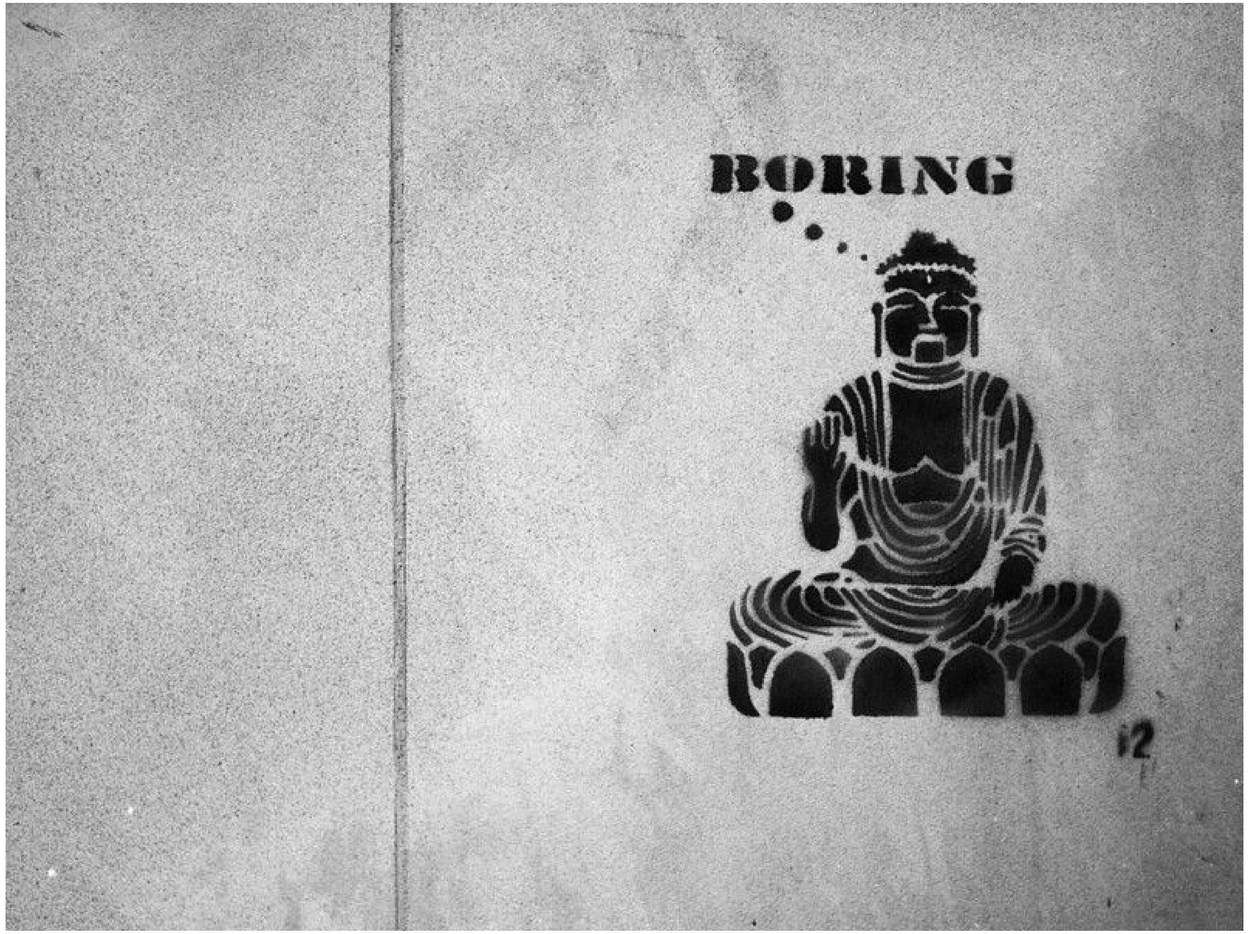
Didn’t change the UX

Didn’t change the data

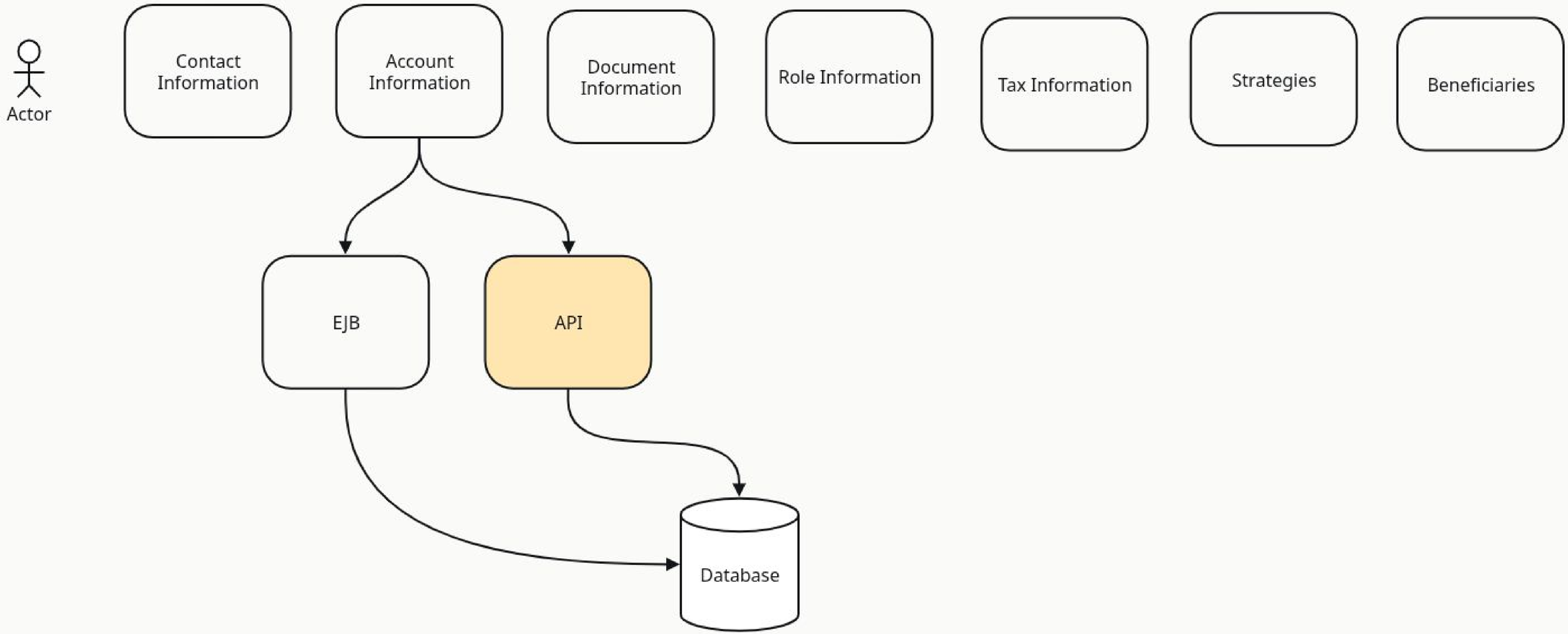
Same data, one type

Working Inside Out

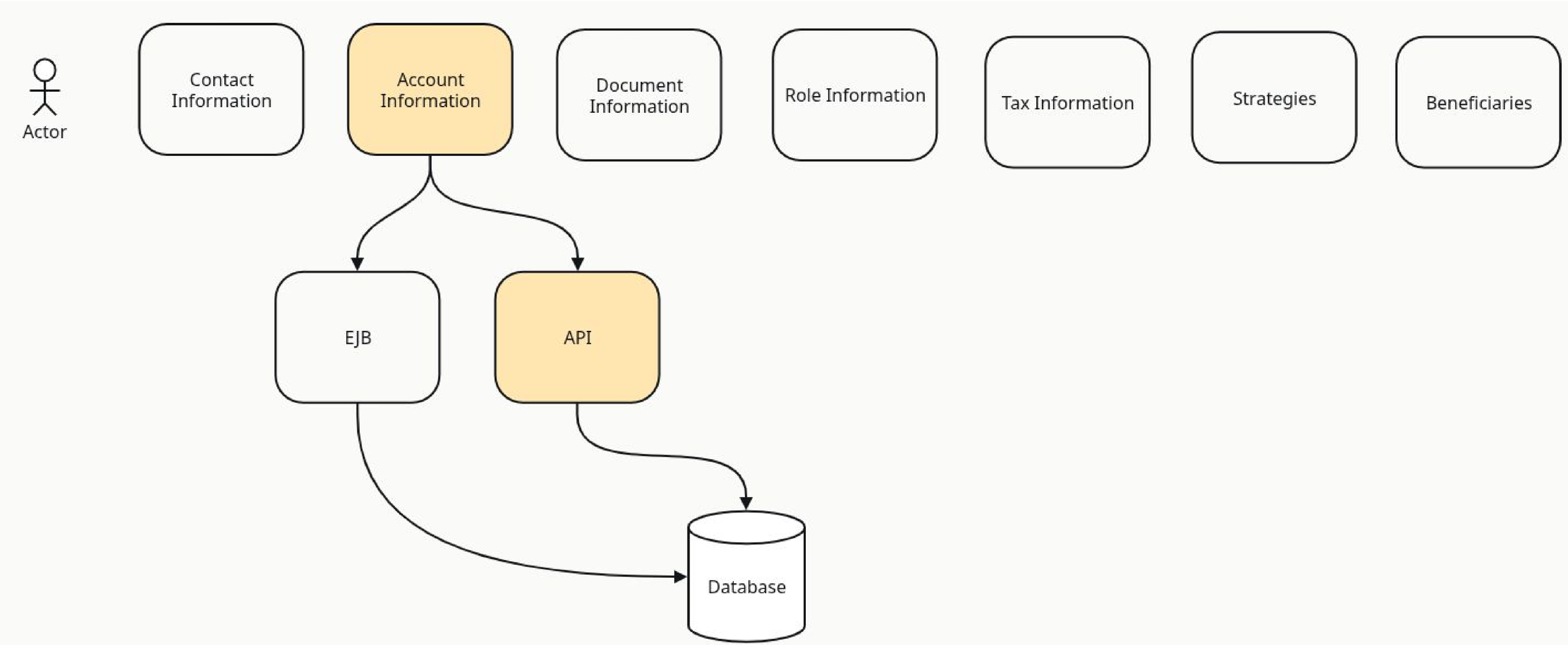
In Mem database (not all the data)



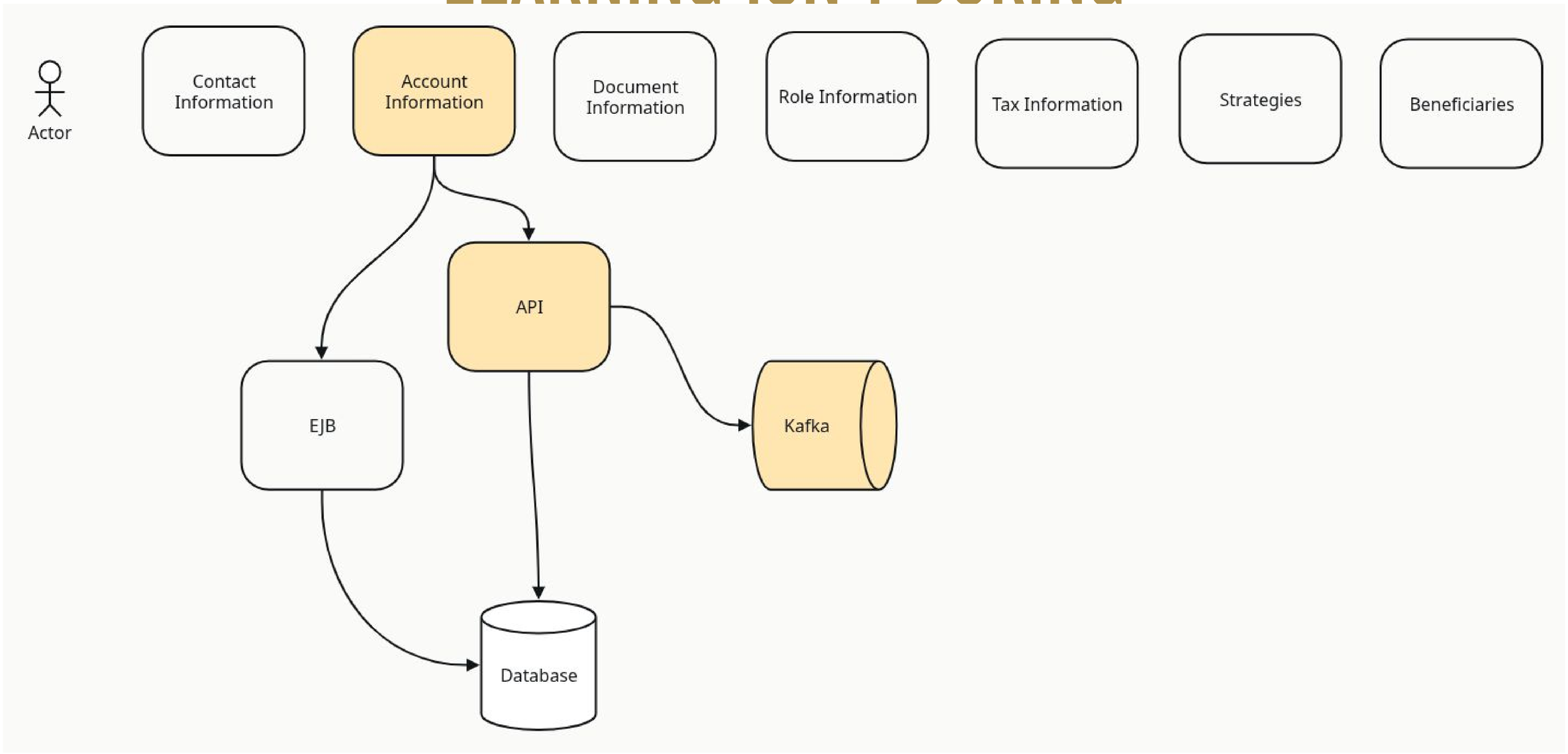
LEARNING ISN'T BORING



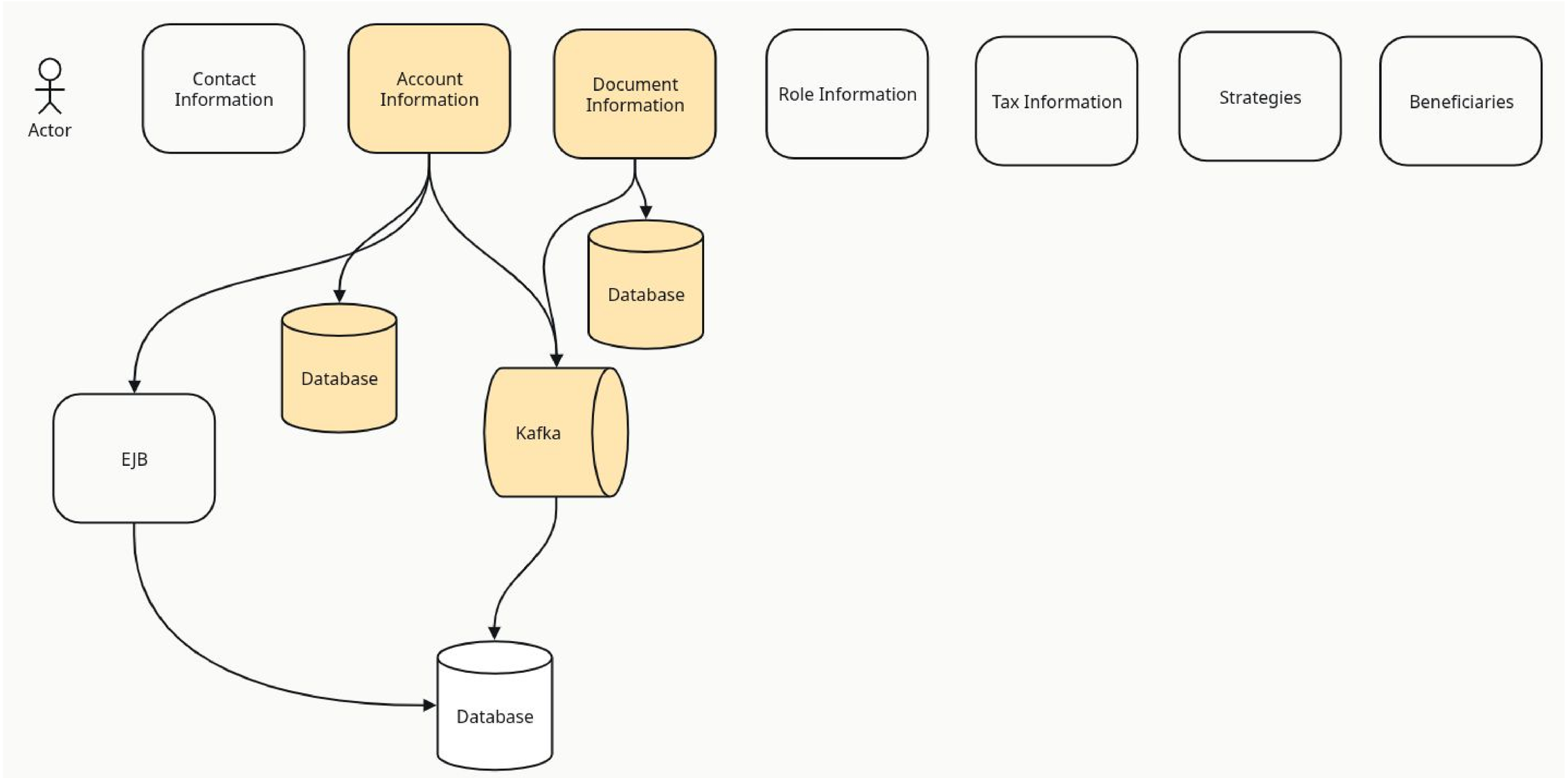
LEARNING ISN'T BORING



LEARNING ISN'T BORING



LEARNING ISN'T BORING



NATURAL EXPANSION AND CONTRACTION
I.E.
DON'T BE SMARTER TODAY THAN WE NEED TO BE

ACTUALLY REFACTOR

Fight the urge to pre-optimize

Focus on boring

Let the code grow

Refactor when you see the patterns, not before

Tests help

ONE APPROACH

- Seam
- Hardcode
- Expand

SHOULD WE MODERNIZE OR MIGRATE? (FIRST BAD MATH)

LOOKING ONLY AT LABOR

Teams have a burn rate of $$$

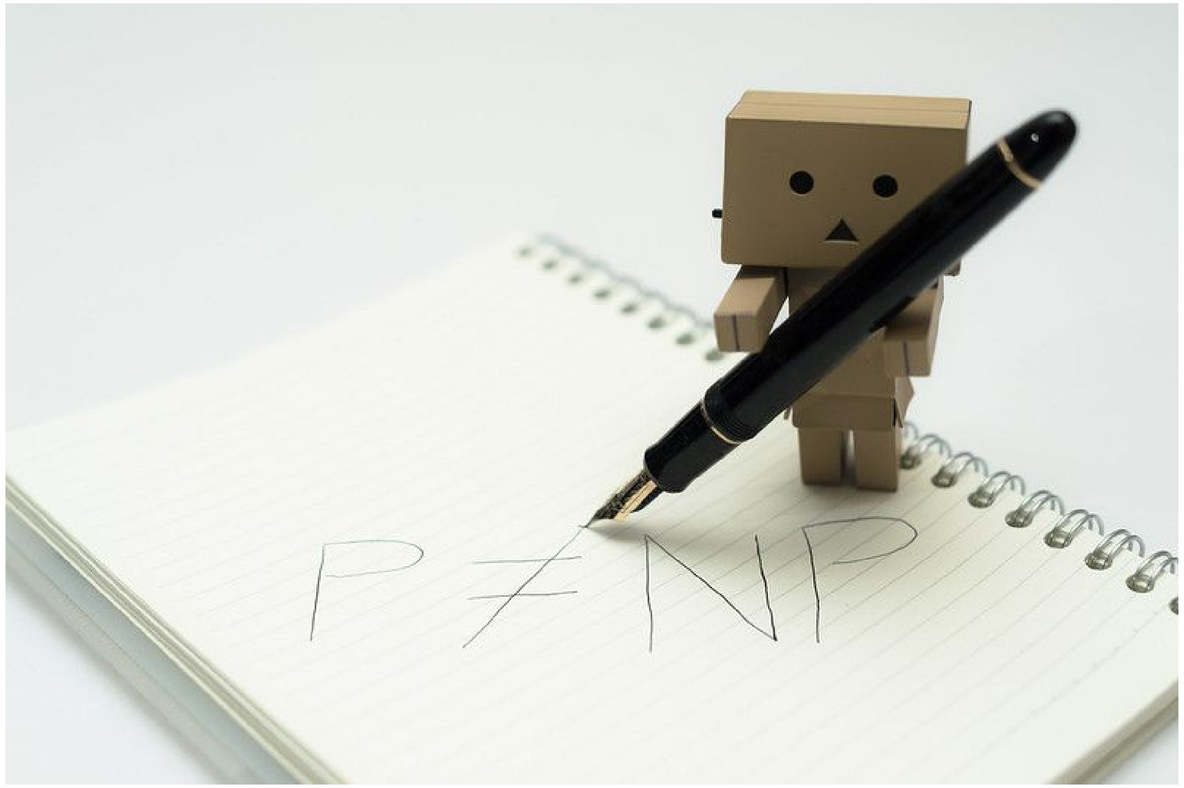
Teams velocity is XXX

Total effort is YYY for all LOC (and the lies begin)

Therefore estimate to complete effort is

YYYY/XXXX * $$$

It is just math, right!



LOTS OF GOOD ANSWERS
COUPLE REAL BAD ONES

U-CURVES

Software Delivery is non-linear
Stop looking for the best answer – look for the the whole range of ‘way better answers’

i.e.
The question isn’t ‘Do we stop product delivery to modernize now or just migrate all the code’? That is 2 bad options to a misinformed question

Think about what you are optimizing for

Find the happy middle ground

COST IS NOT HOURLY

"coin flip" by frankieleon is licensed under CC BY 2.0. To view a copy of this license, visit https://creativecommons.org/licenses/by/2.0/?ref=openverse.

When considering the cost of product delivery, focusing on hourly cost /rate is wrong.

Total cost is a function of delivery and opportunity.

Delivery is made up of active time and waiting time, failure demand and value demand

LOOKING AT TOTAL COST

Opportunity cost - benefits of shipping better UX iteratively, sooner; risk

Delivery cost - the work, plus potential reductions in toil / failure demand, cognitive load

Labor Cost (Migrate, Just Math!): Offshore $13.5M; InHouse $36M

Total Cost (Modernize, Understand): Offshore $52.3M; InHouse $19M

Now we can talk strategy....

RESULTS

14 teams - 15-60 minutes of active development / week
Rest of time - debugging, building, testing

One Year Later

4 teams able to independently deliver features daily
900 line methods down to 50

Continuously speed up

Prod Not Impacted

YOU CAN DO IT TOO
THESE TECHNIQUES ARE EASY
COMPLIMENT EACH OTHER

WHAT QUESTIONS DO YOU HAVE?

Joel Tosi

Slides – https://www.dojoandco.com/speaking

@joeltosi@mastodon.social

Joel.Tosi@dojoandco.com