

# Using DDD as a crutch

 fit x RUG::B

*Berlin 2019-01-03*



# Problem: the glorified ball of mud

- Unbounded legacy systems
- Whack-a-mole issues
- Only tribal knowledge and heroics - speaking all languages at once - save the system from complete collapse



“You reach for the banana and get the entire gorilla”

Domain-Driven

DESIGN

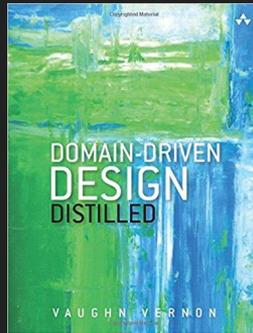
Tackling Complexity in the Heart of Software



dat Kandinsky artwork tho

Eric Evans

Foreword by Martin Fowler



BOUNDED CONTEXT

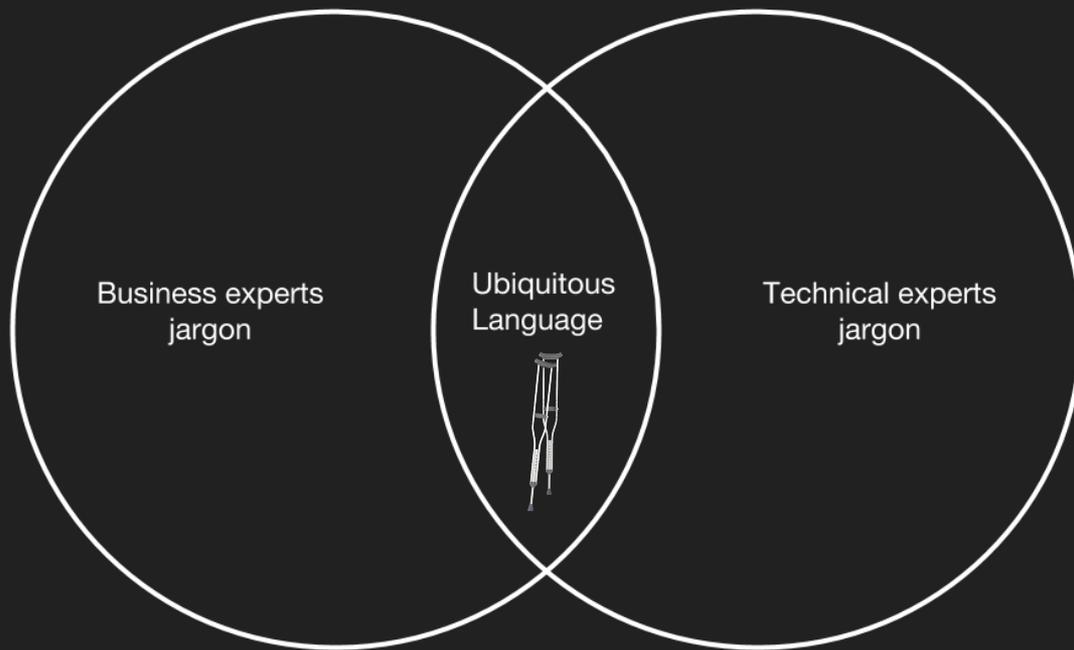
+

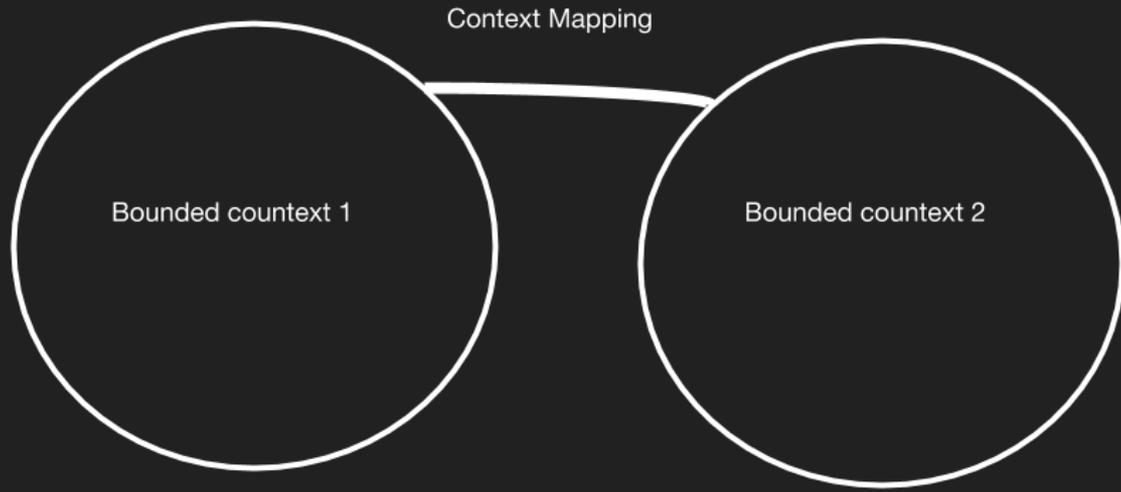
UBIQUITOUS LANGUAGE



# Solution: use Domain Driven Design as a crutch

“Each class, each method, each variable should be carefully named so that the story they tell is the business story you’re writing”





Bounded context 1

Bounded context 2

Context Mapping

# Context mapping has to be strategic

- Anti corruption layer
- Open Host service
- Conformist



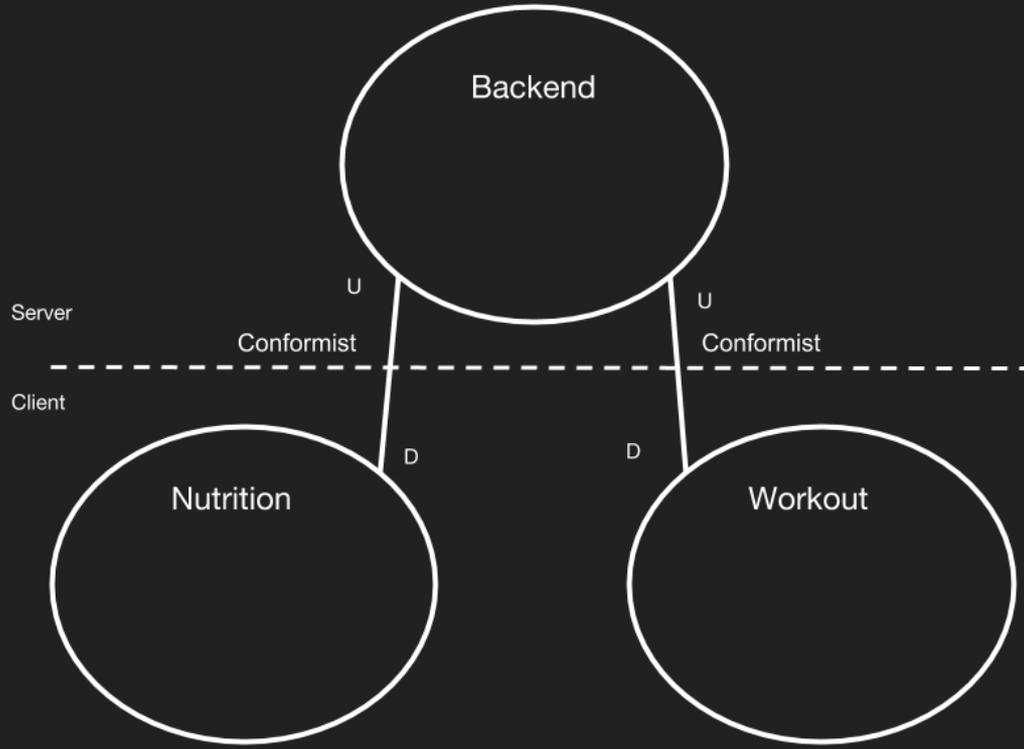




The logo for 8fit features a stylized number '8' on the left, composed of two concentric, rounded lines that form a continuous, intertwined shape. To the right of the '8' is the word 'fit' in a lowercase, sans-serif font. The entire logo is rendered in a vibrant teal or light green color against a solid black background.

8fit

# Current architecture: the Rails way

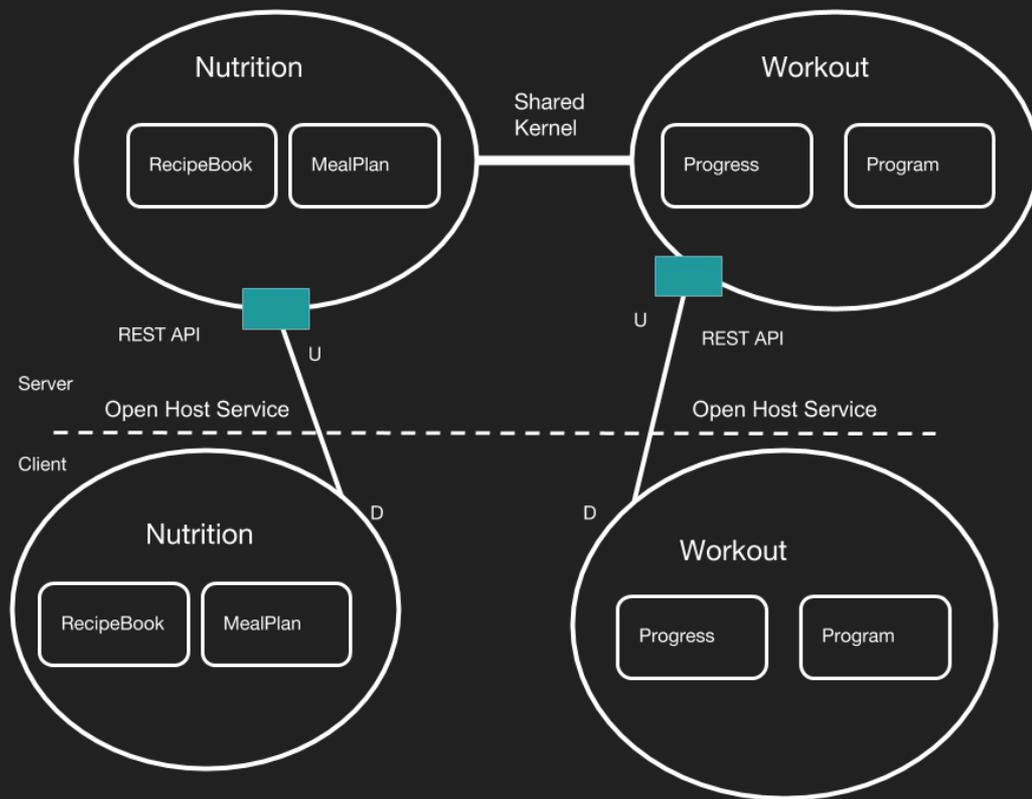


“A common mistake made when using REST is to design resources that directly reflect the Aggregates in the domain model. Doing this forces every client into a Conformist relationship, where if the model changes shape the resources will also. Instead resources should be designed synthetically to follow client-driven use cases.”

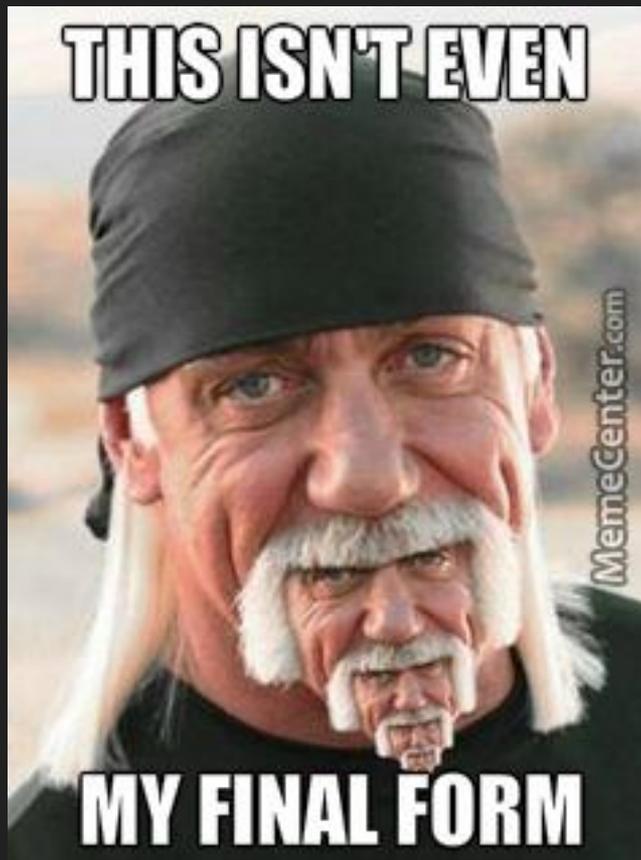
- Vaughn Vernon, Domain-Driven Design Distilled



# Goal architecture: aggregates that reflect business logic



**THIS ISN'T EVEN**



**MY FINAL FORM**

# Overview of the 8fit refactor action plan

## Step 1)

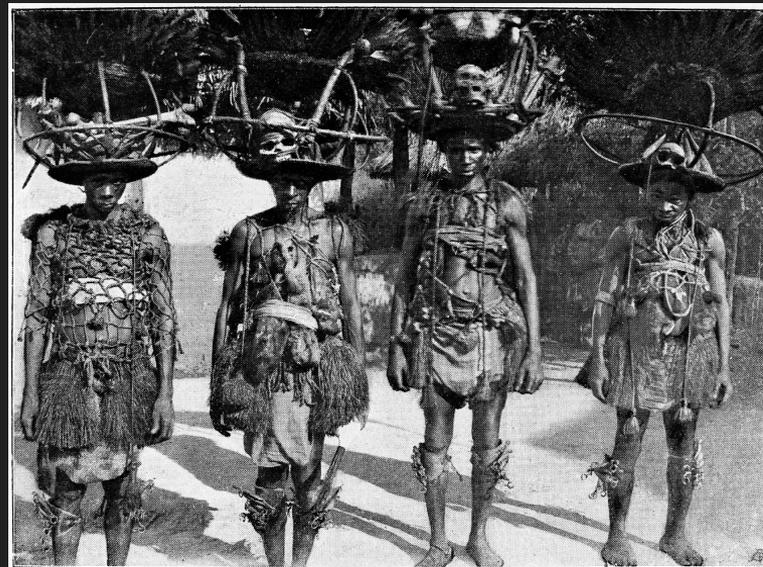
- Isolate bounded contexts into PORO libraries with a shared kernel
- Introduce Open Host Service between upstream and downstream bounded contexts
- Introduce Pub/Sub channels

## Step 2)

- Move more and more towards Event sourcing architecture with the content being stored in a pg database and all the user data being a list of versioned events

# Tools in the Ruby ecosystem

- POROs !
- Wisper
- Rails event store
- ROM-rb
- Dry-rb
- frameworks



# Additional resources

- “Domain-Driven Design: Tackling Complexity in the Heart of Software” Eric Evans (the founding book)
- “Domain-Driven distilled” Vaughn Vernon (tl;dr of the one above)
- Arkency
- Trailblazer/Sequent.io/Component based Rails architecture
- The transcript of this talk (more or less):  
<http://pascal.cc/blog/domain-driven-design>
- Norman Rosner hosting a permanent Q&A on Twitter @normanrosner



Danke

Q&A

or similar stories to share?