

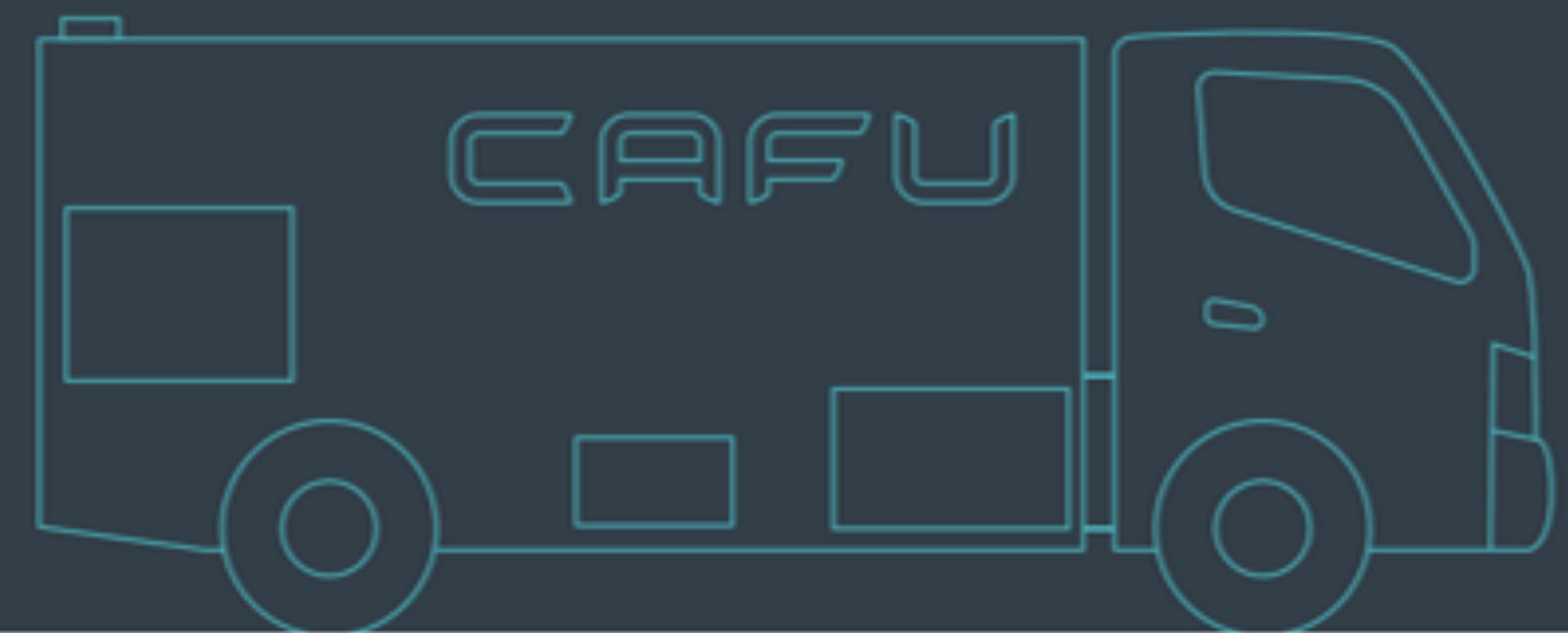
CAFU

# Architecture Homicide

**And other jokes you can waste your emotional energy on**

Ahmed Ramy - iOS Software Engineer @ [CAFU](#)

# Disclaimer





# So What's the Plan, Cap'ain?

## 1. The Problem

1. Suspect #1: Silver Bullet (5 mins)
2. Suspect #2: It's notoriously famous and must be used (10 mins)
3. Suspect #3: Consistency is the key towards a better world (10 mins)
4. Suspect #4: Wrong Tool @ Wrong Context (5 mins)
5. Suspect #5: My Emotions are too strong and this "feels" so wrong (3 mins)

## 2. The Results? (5 min)

## 3. The Otherside? (10 mins)

## 4. The Important Questions

1. Why?
2. How?
3. When?

## 5. Questions

## 6. Quick Live-Designing

1. Authentication w/ Multiple Providers
2. Strategy Pattern
3. Diagrams
4. Testability

## 7. Reasons of Change

**Let me tell you a little story about a troubled  
craftsman**



The weekend ends, the weekdays start, you arrive at the office...

Yet another bug, yet another patch here, a patch there and lots of crappy, iterative, duplicated routines you have to make 😢

You, somehow, managed to survive yet another week

**UNTIL**, the inevitable happens...

The deadline hits ya hard...



**Brain mode gets activated**

**You begin scrambling any library you want off Github...**

**You sew a couple of different solutions together, force unwraps are all over the place**

**Any scenario outside the happy path is unwelcome**

**And that's where the rise of crash rates begin**

**And the fall of the Ratings begin**



And once you take a breath, you begin blaming others

It goes between the lines of "**MANAGEMENT WHO JUST CAN'T UNDERSTAND!**"

Or maybe

**"MARKETING WHO JUST HYPED US TOO BIG"**

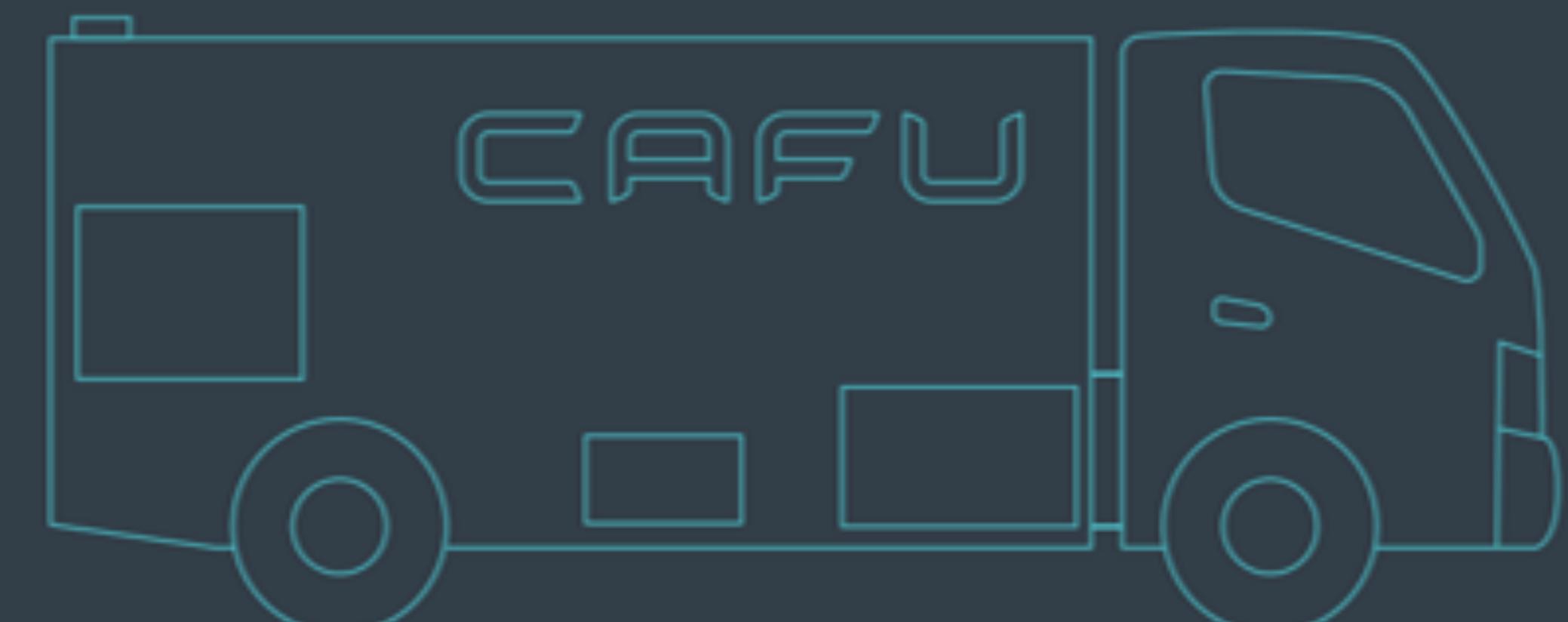
Or

**"{Placeholder for any excuse you want here}"**

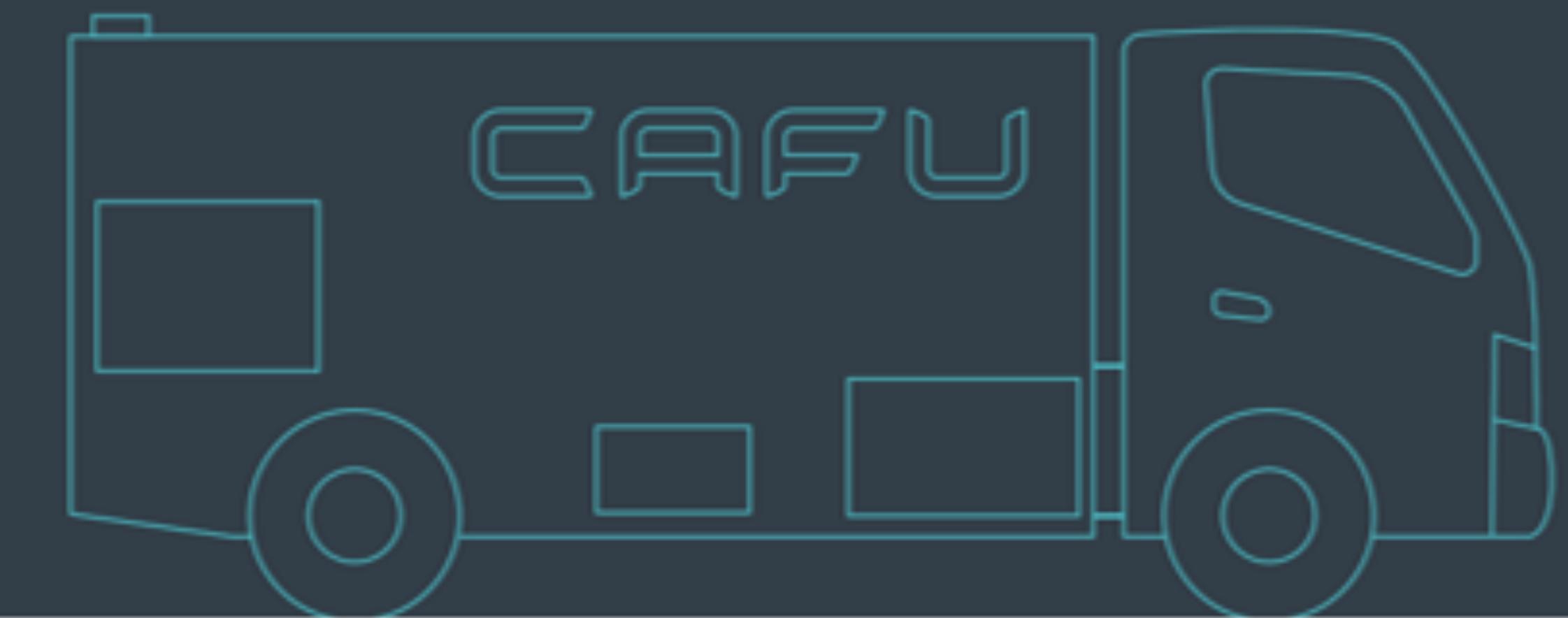


**Sounds  
familiar, right?**

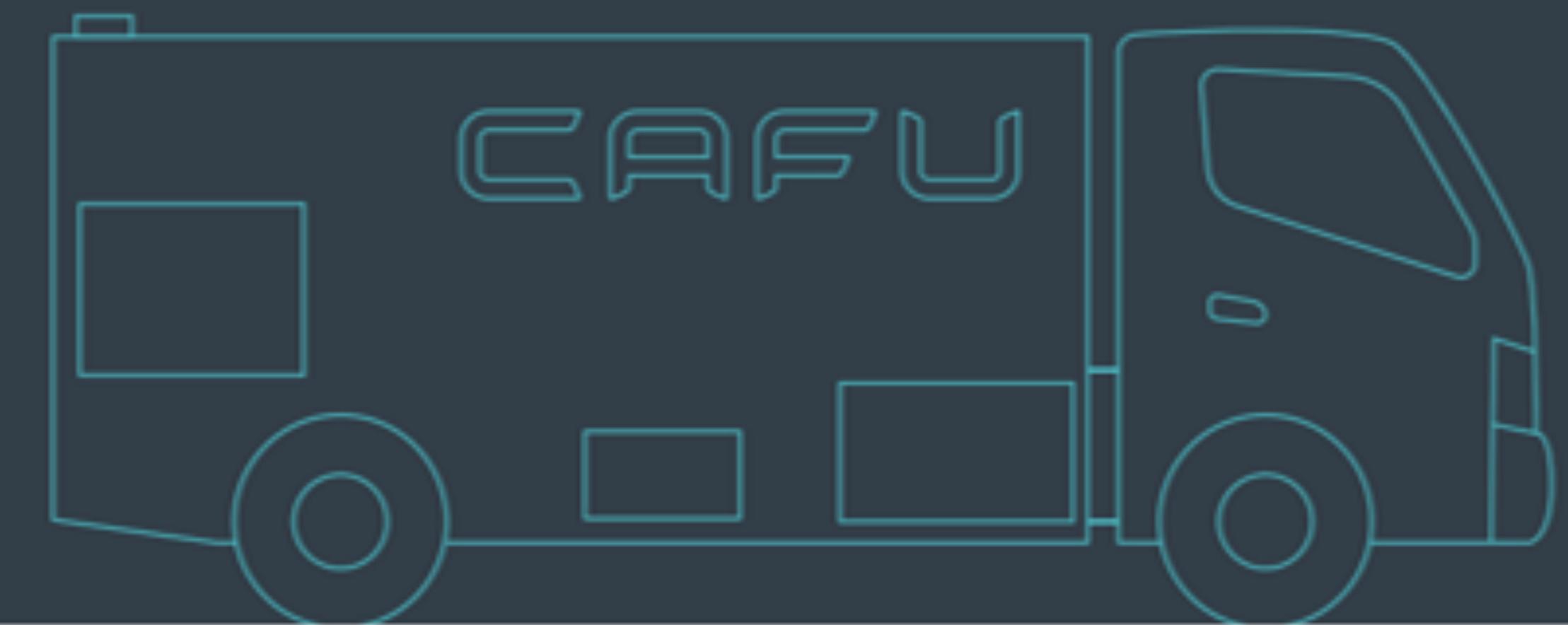
**(Or not yet that is...)**



There must be a better way! 😡



And there is



# Me9uid

## Man of the year



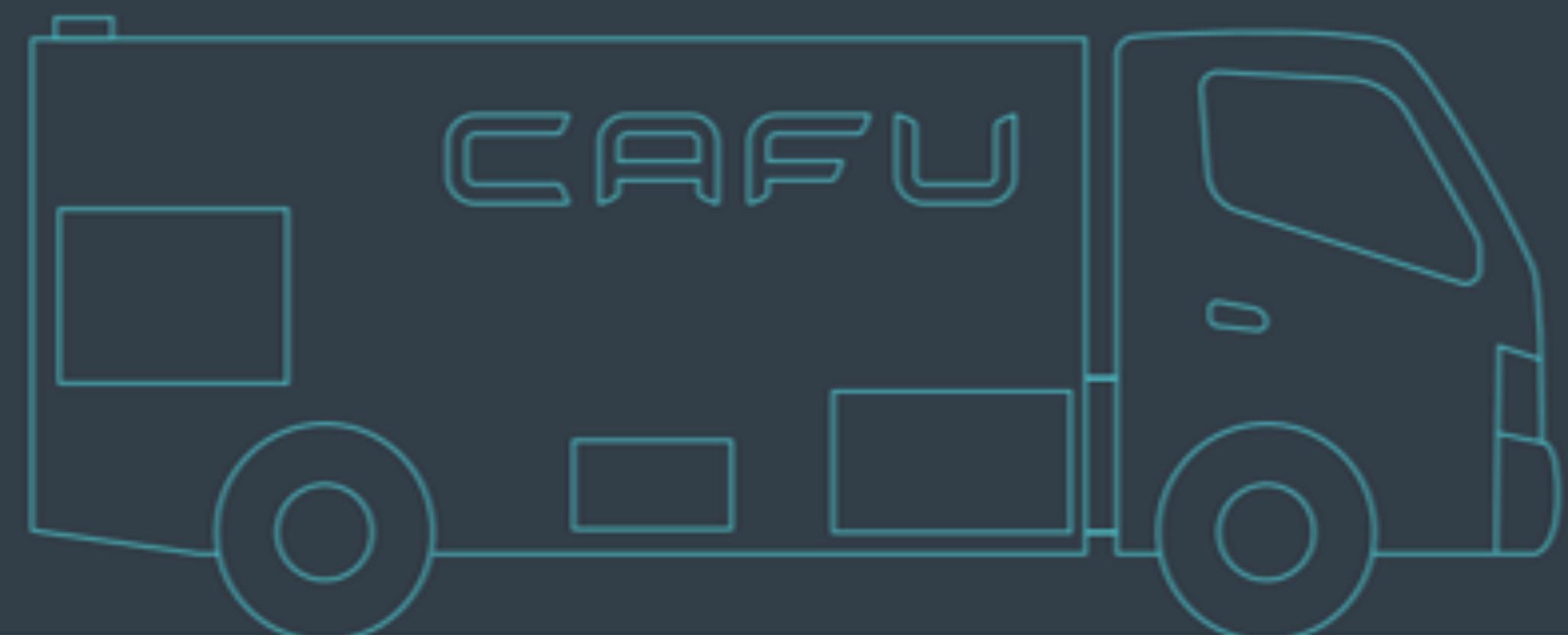
# Crappy Uber Code to refactor



# Quote Number 1



Software Design is all  
about knowing  
`Where` to put `What`



# Question

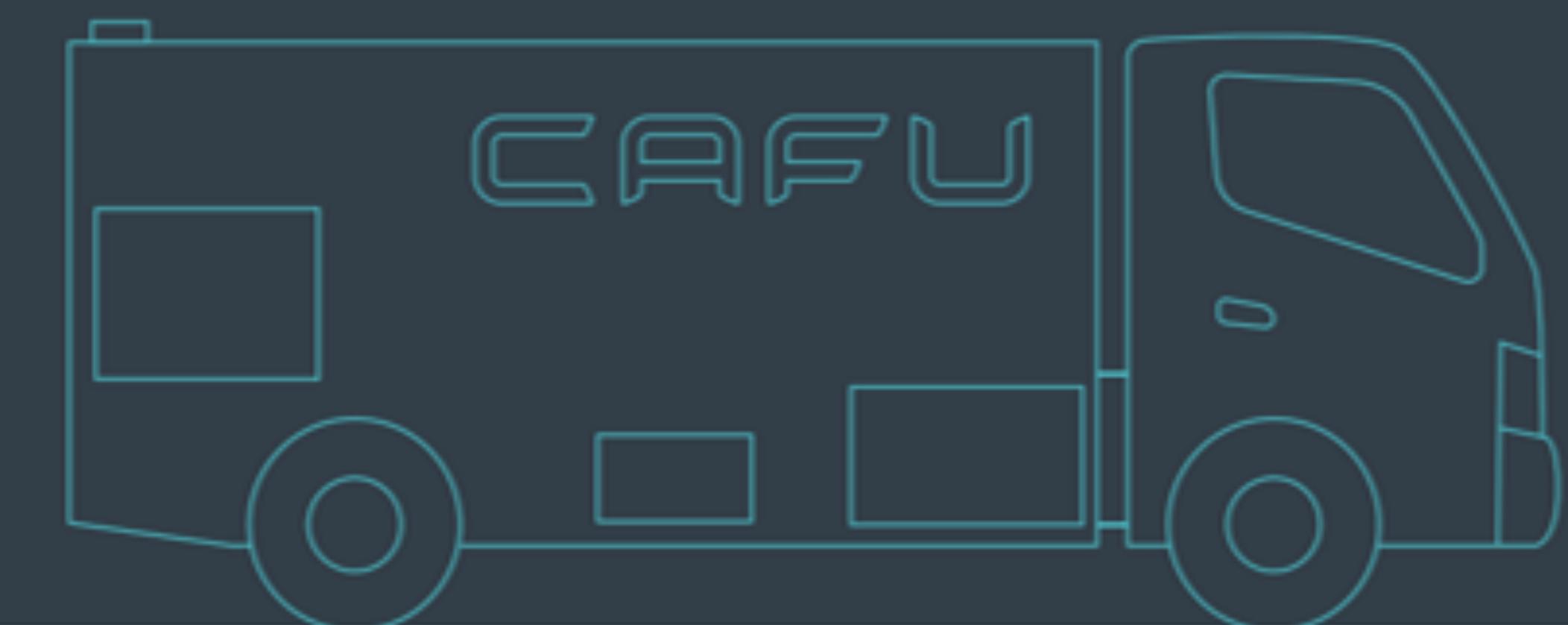
Why do you think big names in the global iOS Community can juggle jobs quickly?  
Working as Backends, Frontends, build their whole websites even tho they are professionally committed to iOS?



They Get Comfortable  
with the most  
uncomfortable topic ever



# The Art of Software Design



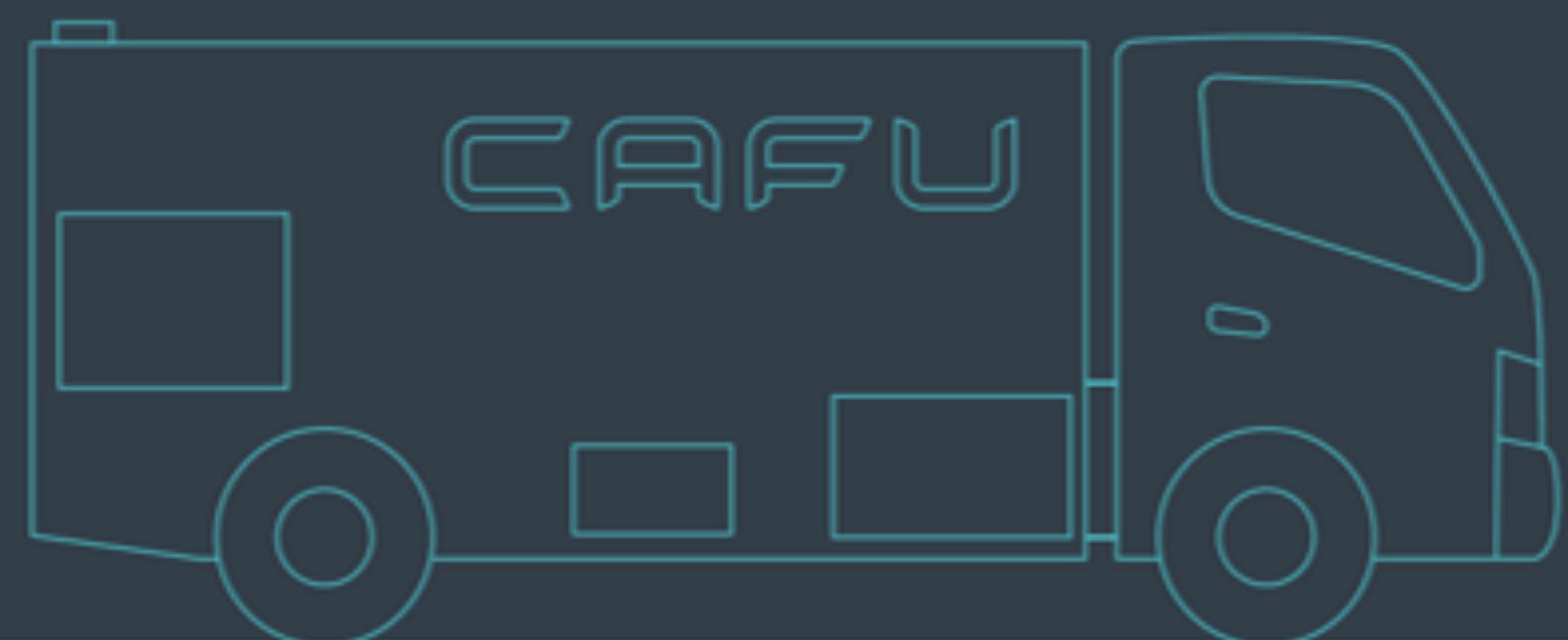
# Quote Number 2



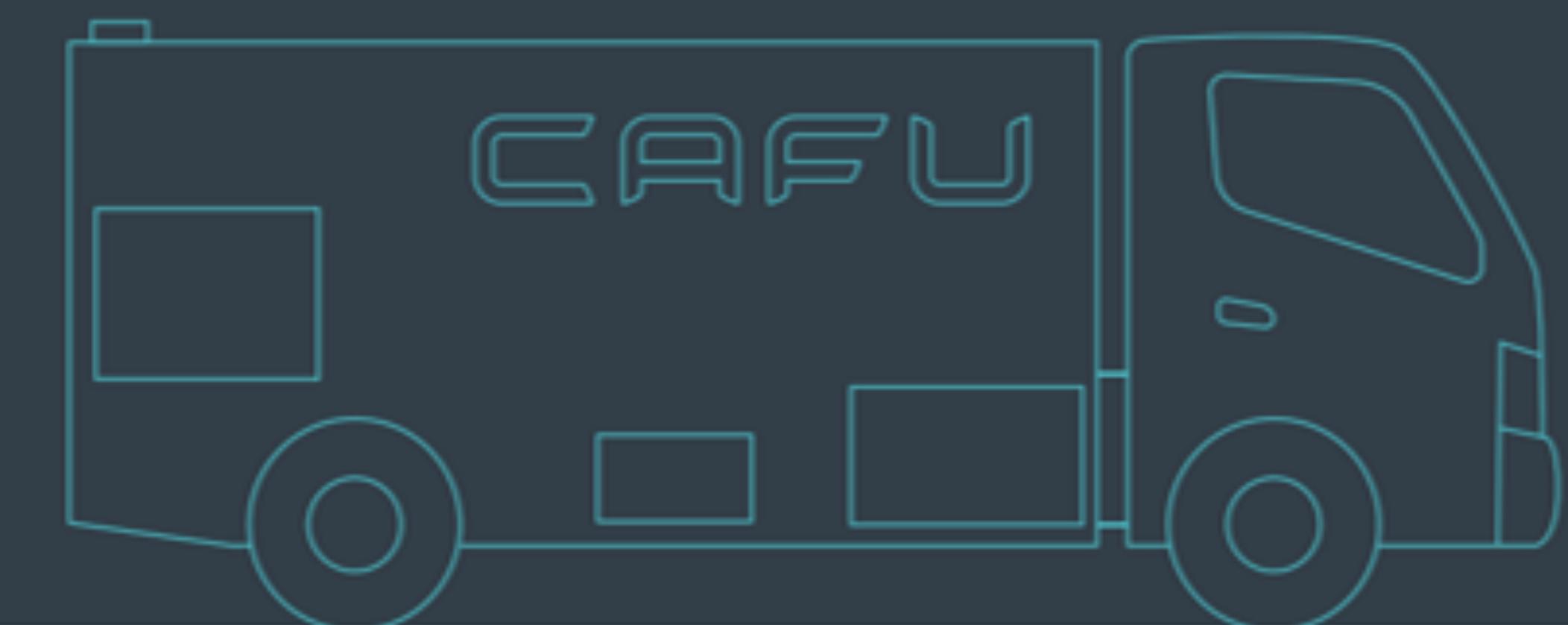
**Software Design  
Makes Up 95% of  
what we do**



So, If you do  
the math



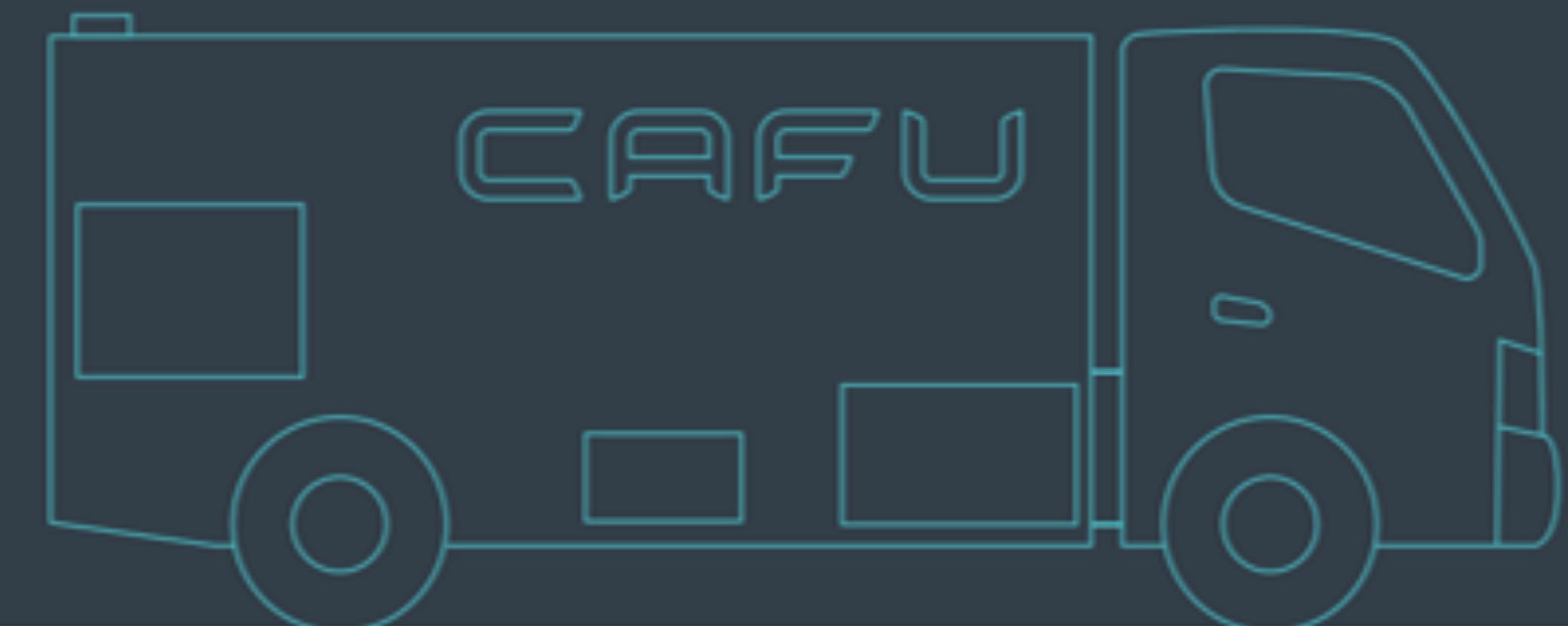
The rest is just  
**Syntax**



And from there, the story of learning,  
observing and bettering, had began



# Why?

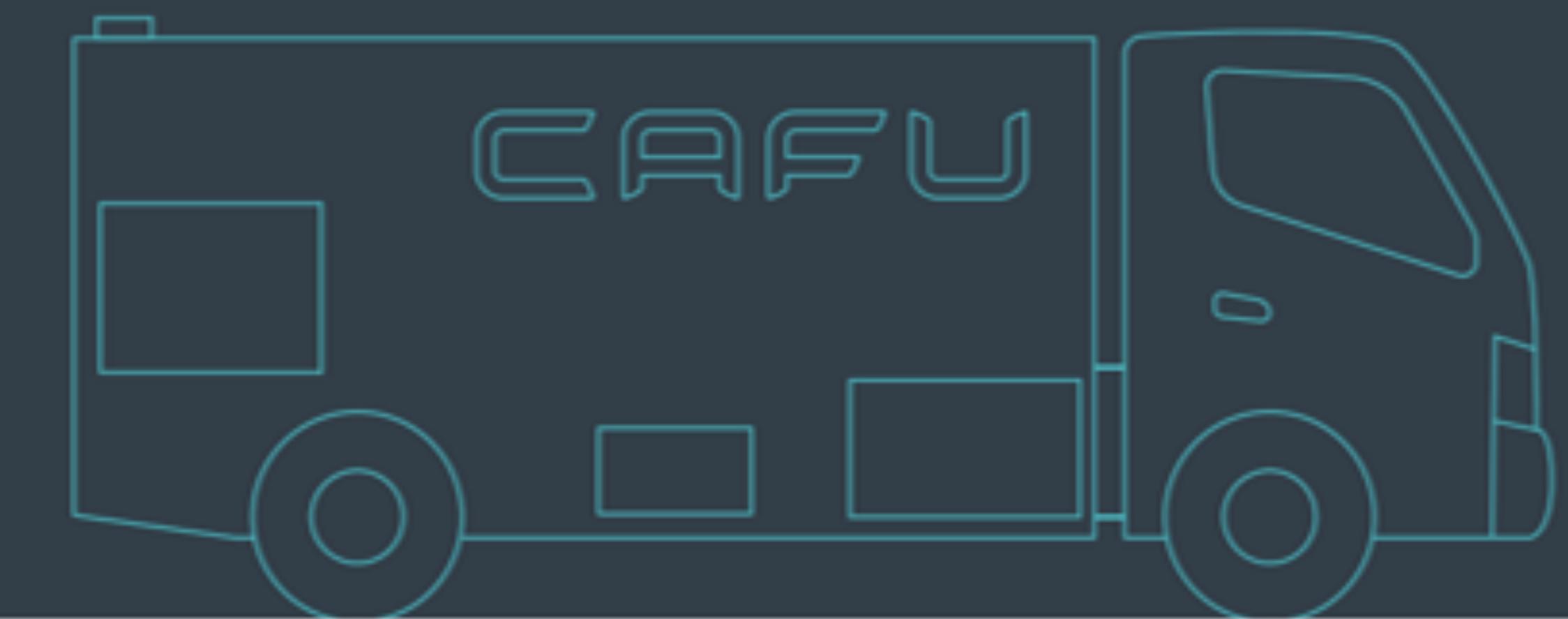




**In the early days in my career, even now or probably forever  
(I think Uncle Bob talked about it in his time as well)**

**you will often see great tournaments of who holds the  
championship of the “Greatest Architecture of all”**

# Something like this





MVC

Model

MVVM

MVVM?

View

VIP?

VIPER?

RIBs?

Massive ViewController

**But is there such thing? And even  
tho, why would it be so great?**



**Some people would say for binding...**

**for ease of use...**

**for testing...**

**for separation of concern...**

**but there are some people who would say.....**

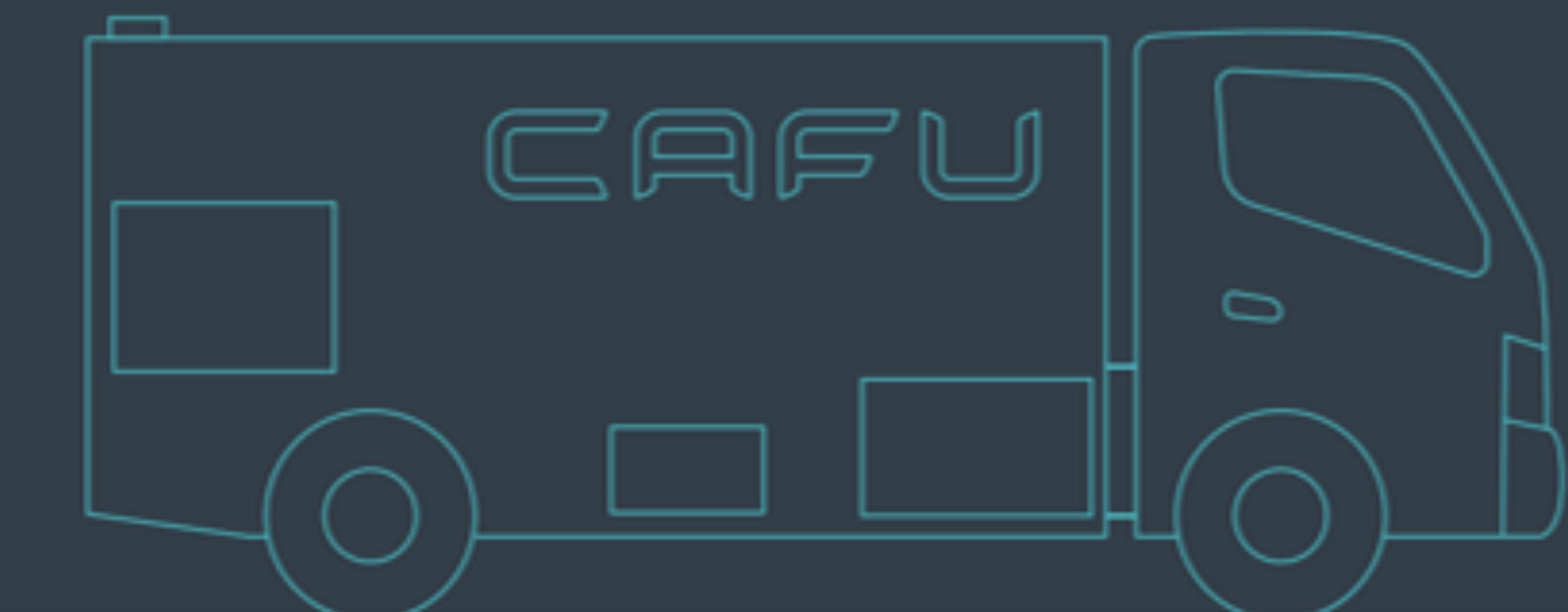




**IT'S THE LAW**



**But, why is that mentality or way  
of thinking is not scalable?**

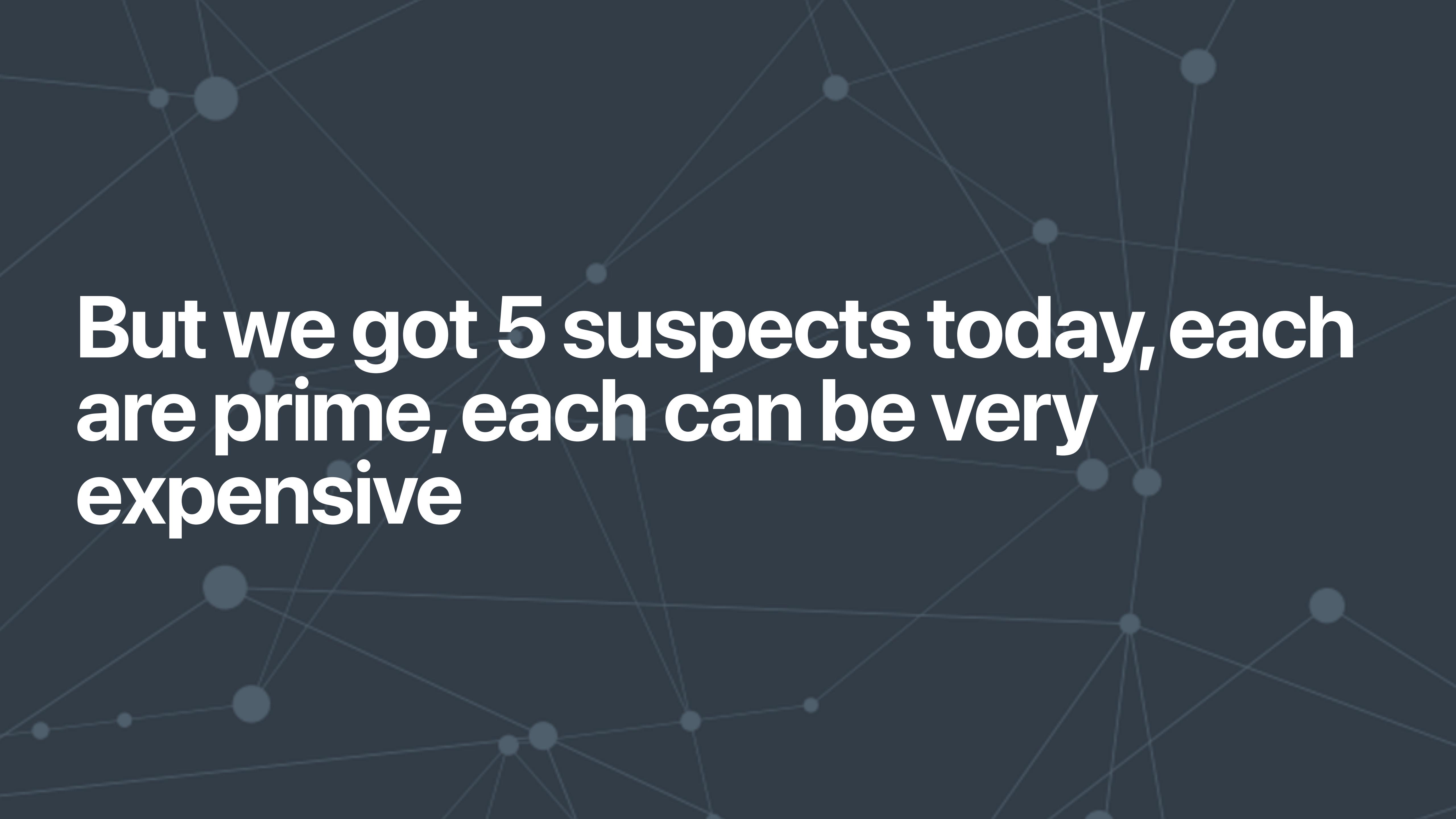


**Deeming something as the greatest or the perfect  
in a non-perfect world coming from the fingertips  
of a non-perfect being is a not-so-perfect claim...**



There are no right or wrong  
answers here, only expensive ones



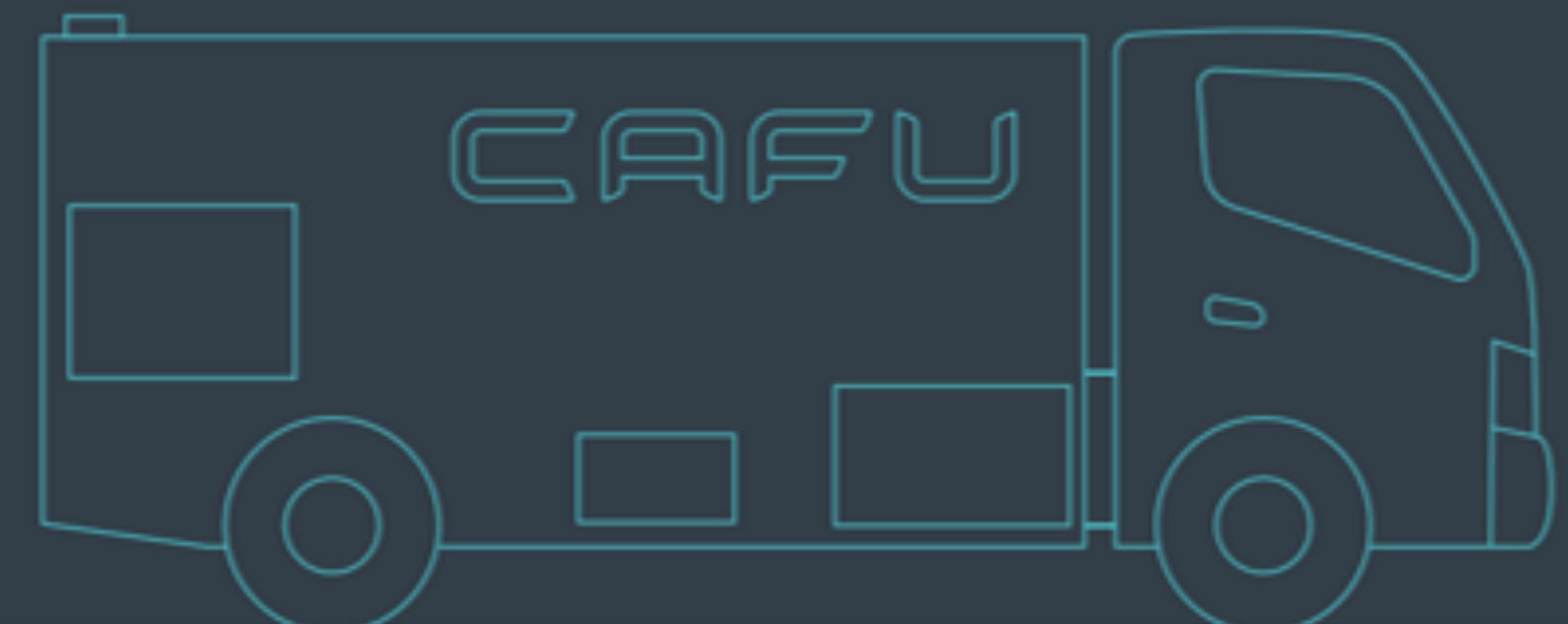


**But we got 5 suspects today, each  
are prime, each can be very  
expensive**

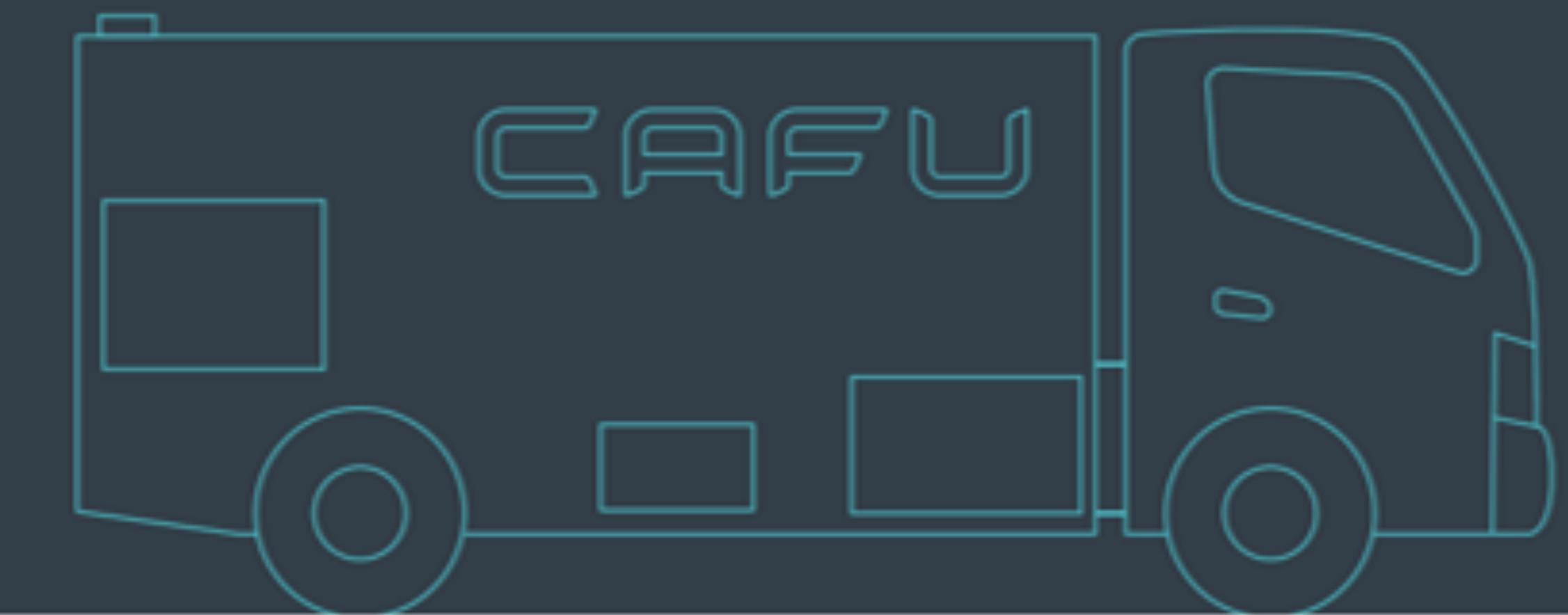


Mr. "Hold my Beer", Where's my  
Golden Hammer?

**Golden Hammer, is another word for Silver Bullet, only difference  
is...**



It's not.



**When you want to keep using a certain technique, paradigm, technology, or a pattern, even when it doesn't make sense in some cases, but it sure is comforting to use**





# Amal Khalil, My First CTO from elmenus



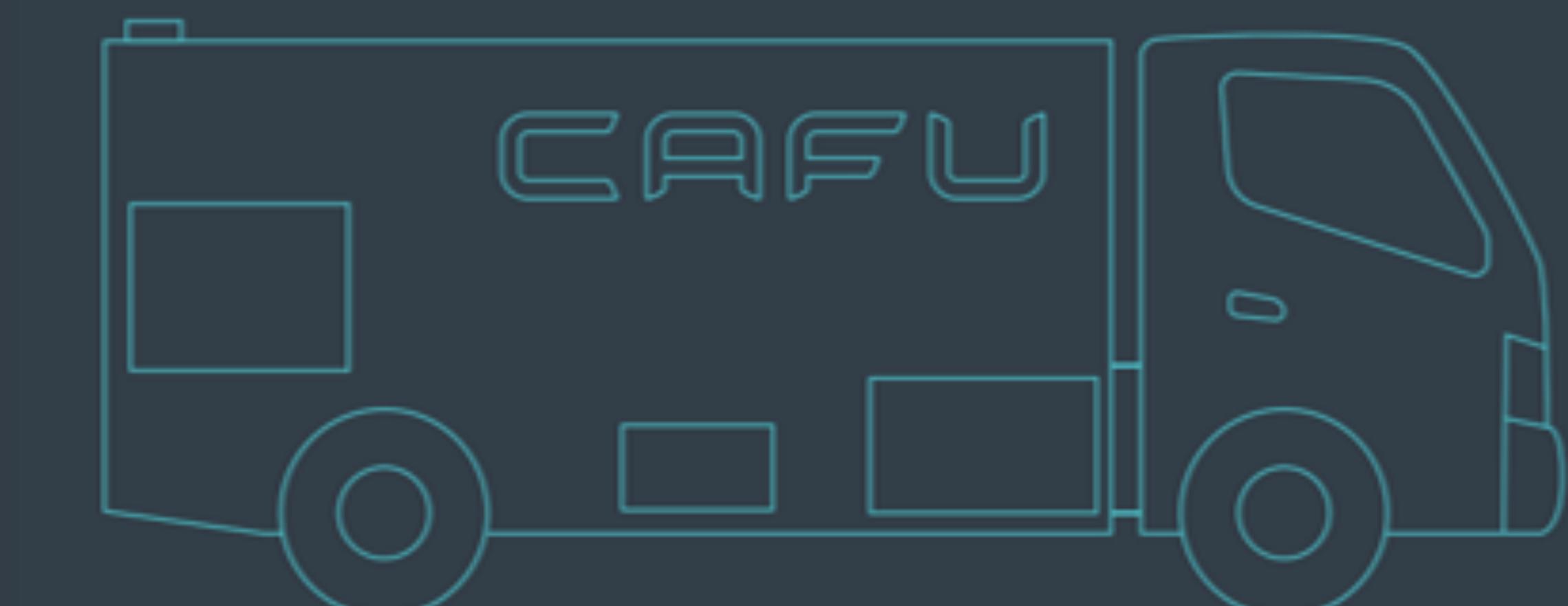
# The consistency rabbit-hole

**The same goes for strictly applying VIPER to a Terms & Conditions Screen where you just throw a bunch of text with some System font just for the sake of consistency and not getting confused (... in a terms & conditions page? I hardly believe that)**



# Mr. Wrong-Tool-in-the-Wrong- Context

**The same goes for strictly applying MVP with no KVO,  
Observables, Publishers or any dynamic way of flowing data in a  
social app or “Real-Time app”, just because MVP doesn’t work  
that way and we have to follow the LAW!**





The “**That doesn’t feel right**”  
sensation and I’ll throw in your  
face any unreasonable excuse

**The same goes for shying out of ReSwift, Redux, and Composable Architecture in an app that has tons of state changes and manipulation just because they're strange...**



**Worst excuse?**  
**This one uses Global functions, this “Bloats the Autocompletion”**





**Mr. Hyped for this Technology!**

**Asking a team who are not that great with OOP to implement a Microservices Backend Architecture using Scala (which is purely Functional) just because it's GREAT will give you the greatest firing in your whole career when the backend falls apart (am just exaggerating here)**



**(but Airbnb wasn't when their CTO decided to go with React Native since it was the Hype back then, was a very expensive decision for him, and Airbnb)**



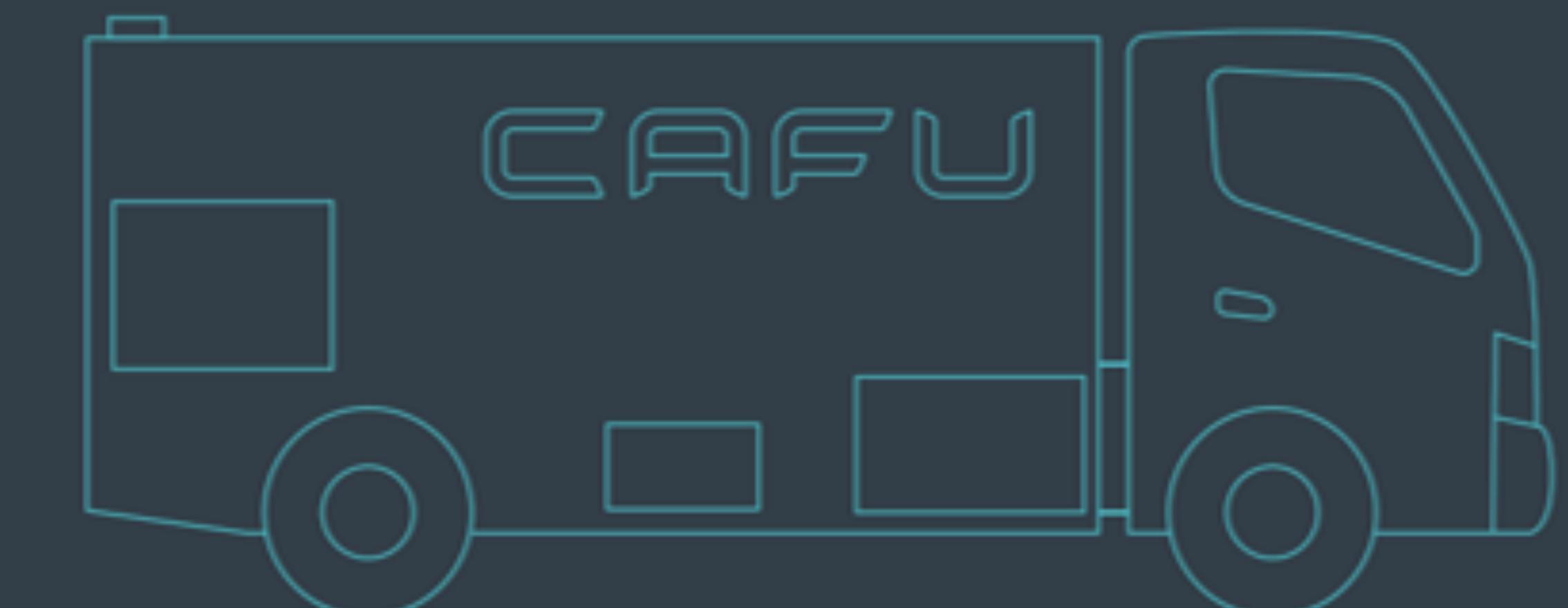
**If you're a decision-maker for what technology, approach, architecture, or pattern to use, don't make ur reasons be “trying this tech. Out”, that's not ethical, and you're trusted to do what's best for the business, not your C.V.**



The same goes in iOS, don't ask people to use RxSwift with MVVM when all you have is just plain old MVC where the ViewController is massive and the architecture is "full of strange things that they call protocols and layers"



**Teach people to learn better first,  
then refine**





The Common between them all?

Relying heavily on Emotions, and  
ending up with a bad structure



**Emotions don't go hand-in-hand with scalability because they are volatile, irrational, and sometimes has no explanation**



# The Hope?



**Instead of using something for **the sake of using it****

(Yep, that's very very famous in us humans until we learn better)

**Teach yourself why was **MVC** made that way, and why **Apple** recommended it**



Then understand the problem that made people shift from MVC to something else



Is it the problem of the **Design**, or  
the **Implementor**?



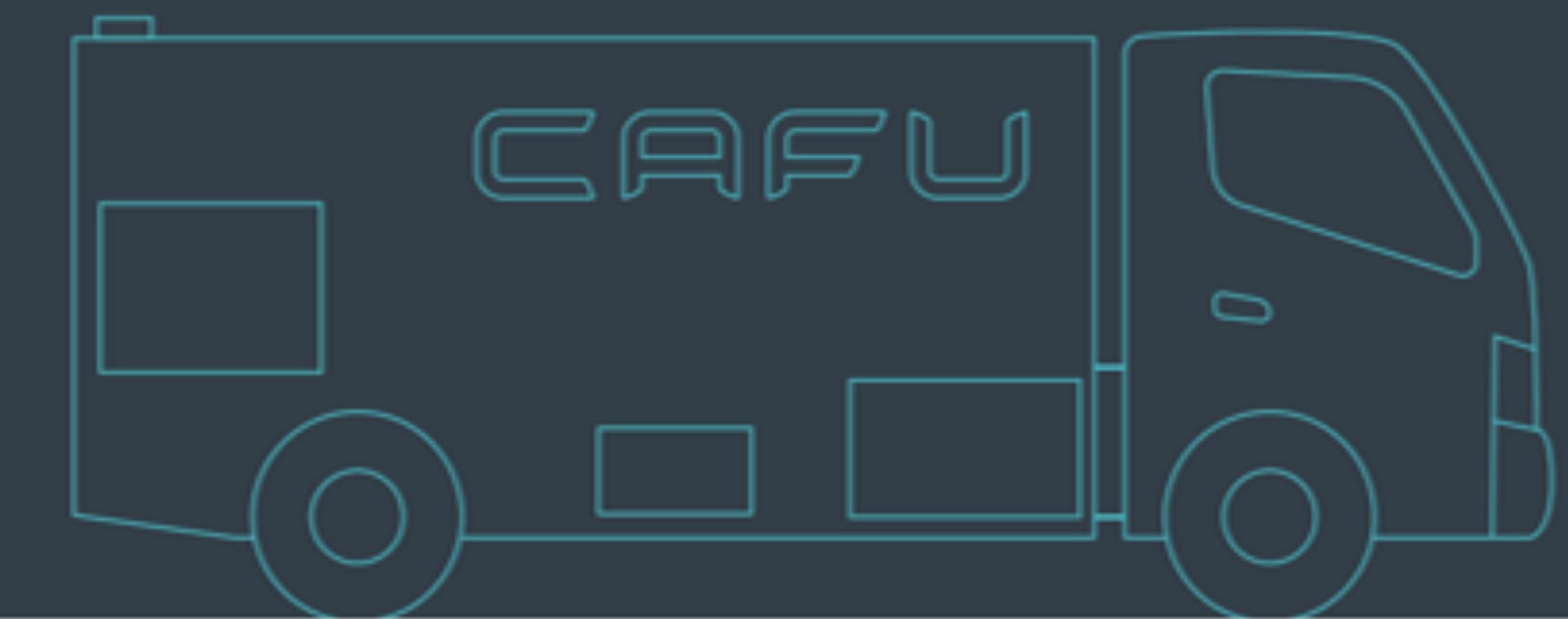
**'Name'** things right  
To be able to put them **'where'** they  
belong.



Then you can go from **2000+ lines**, to a  
**barely <50 lines**, of **maintainable code**,  
**that easily change, easily scales**



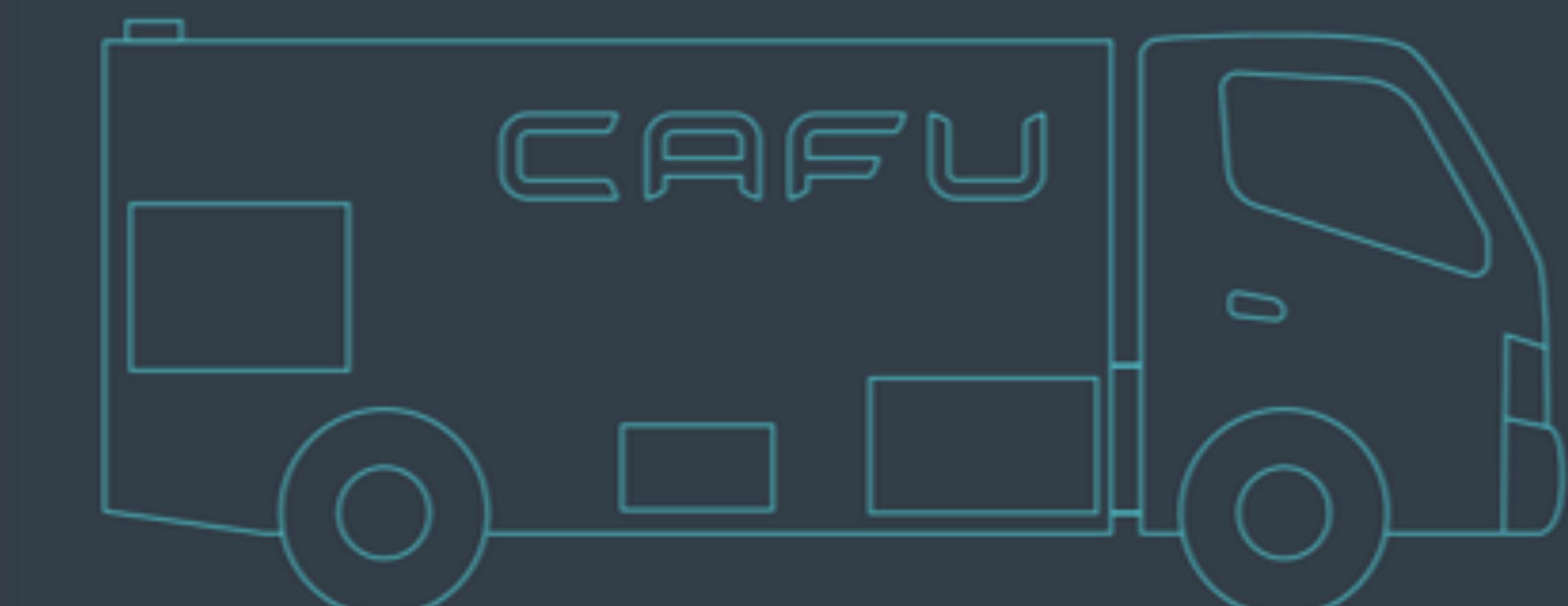
# Story Time



This is not a **fantasy**, nor are we talking bedtime stories of a Utopian career, and no bells will ring to wake you up into a **different reality**



This is good Software Design in Action



**But the question here is, Are we able to do all of  
that absolutely?  
That depends on one thing, is your code testable?**



If your code is testable, your structure is right, if your structure is right, your Design is not messy



Any Questions,  
So far?



# Quick Walkthrough to Designing an Authentication Screen



# Any last Questions?



That's all folks!

