

Rails Meetup 2021

Hoelzle AG
Gregor Wassmann
MSc ETH



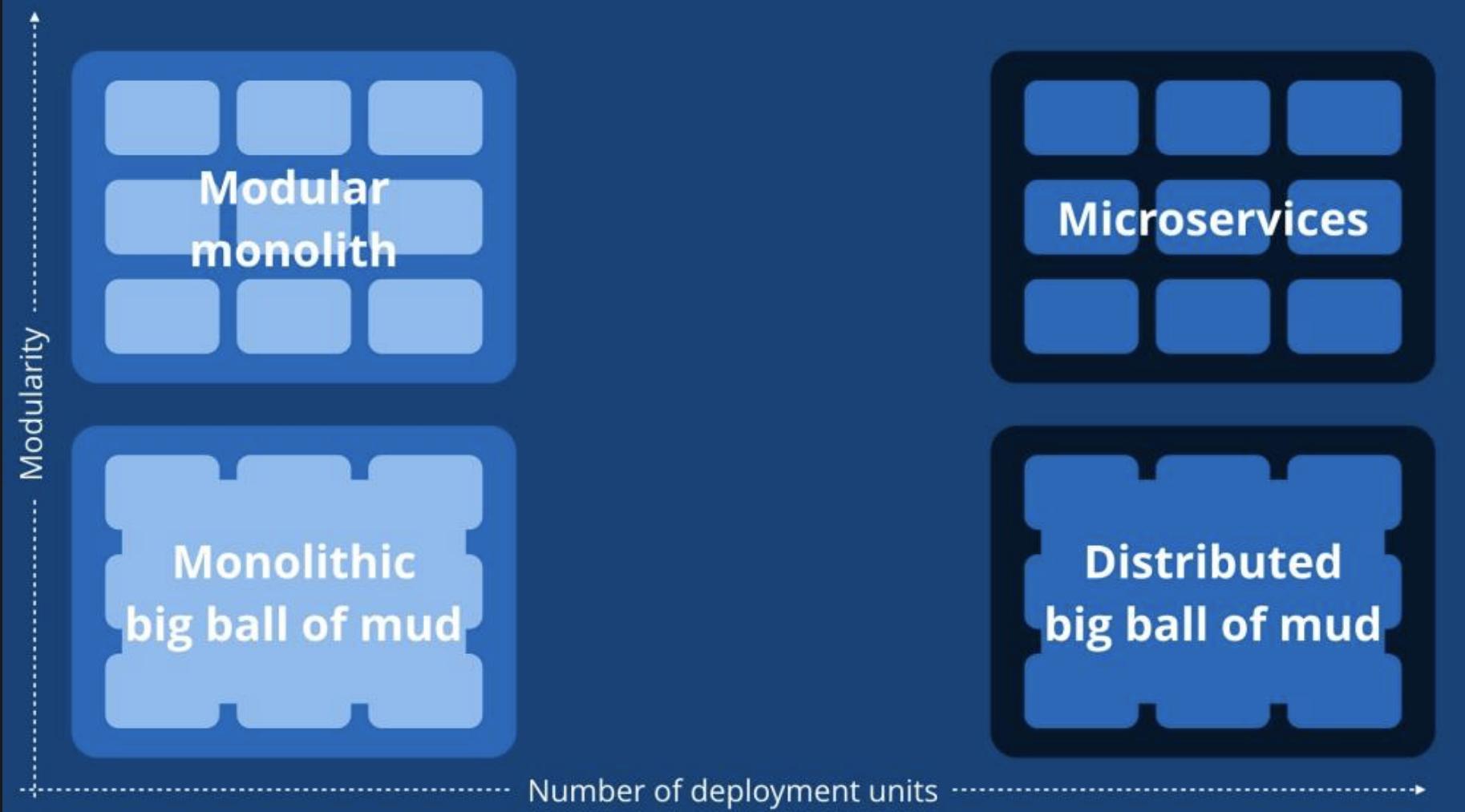




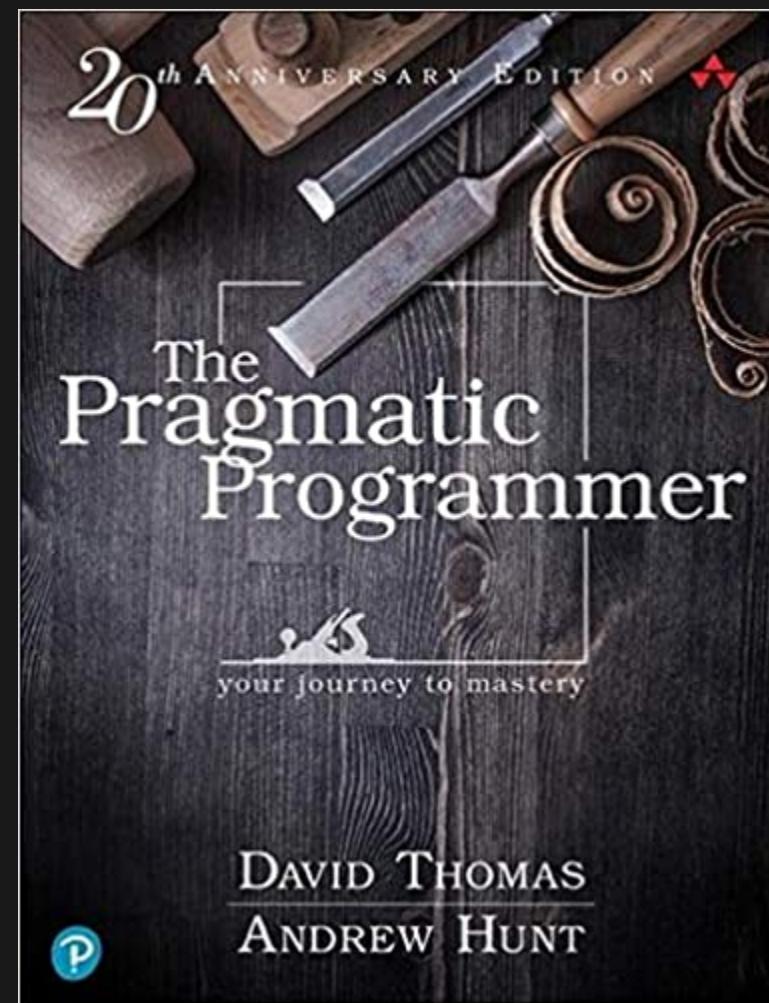
Please find slides and source code on
github.com/hoelzle/rails-meetup-2021

Modular Monolith

Bonus: Best of Rails *and* JavaScript



Entropy



While software development is immune from almost all physical laws, the inexorable increase in entropy hits us hard. Entropy is a term from physics that refers to the amount of “disorder” in a system.

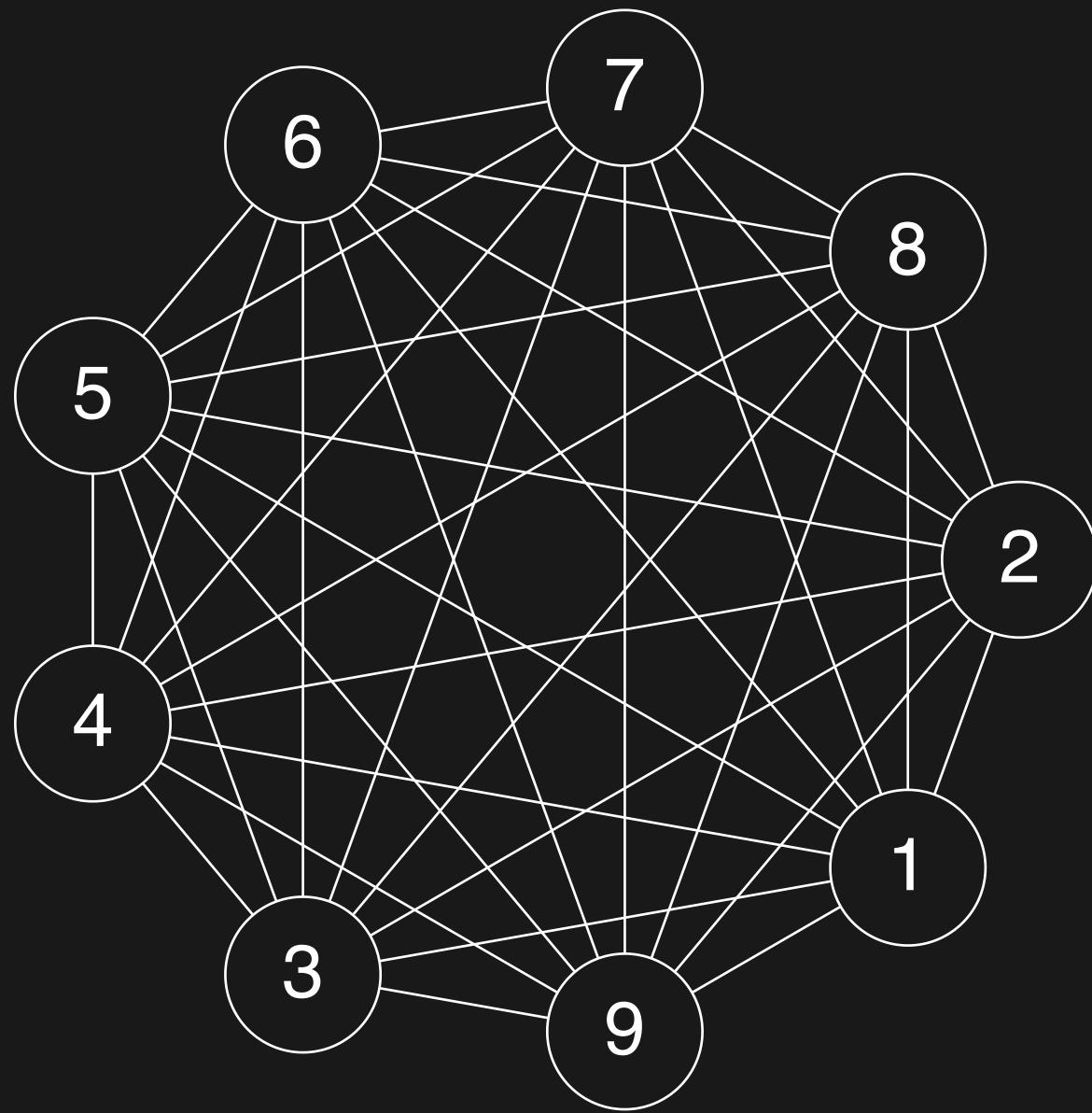
— David Thomas, Andrew Hunt.
The Pragmatic Programmer.

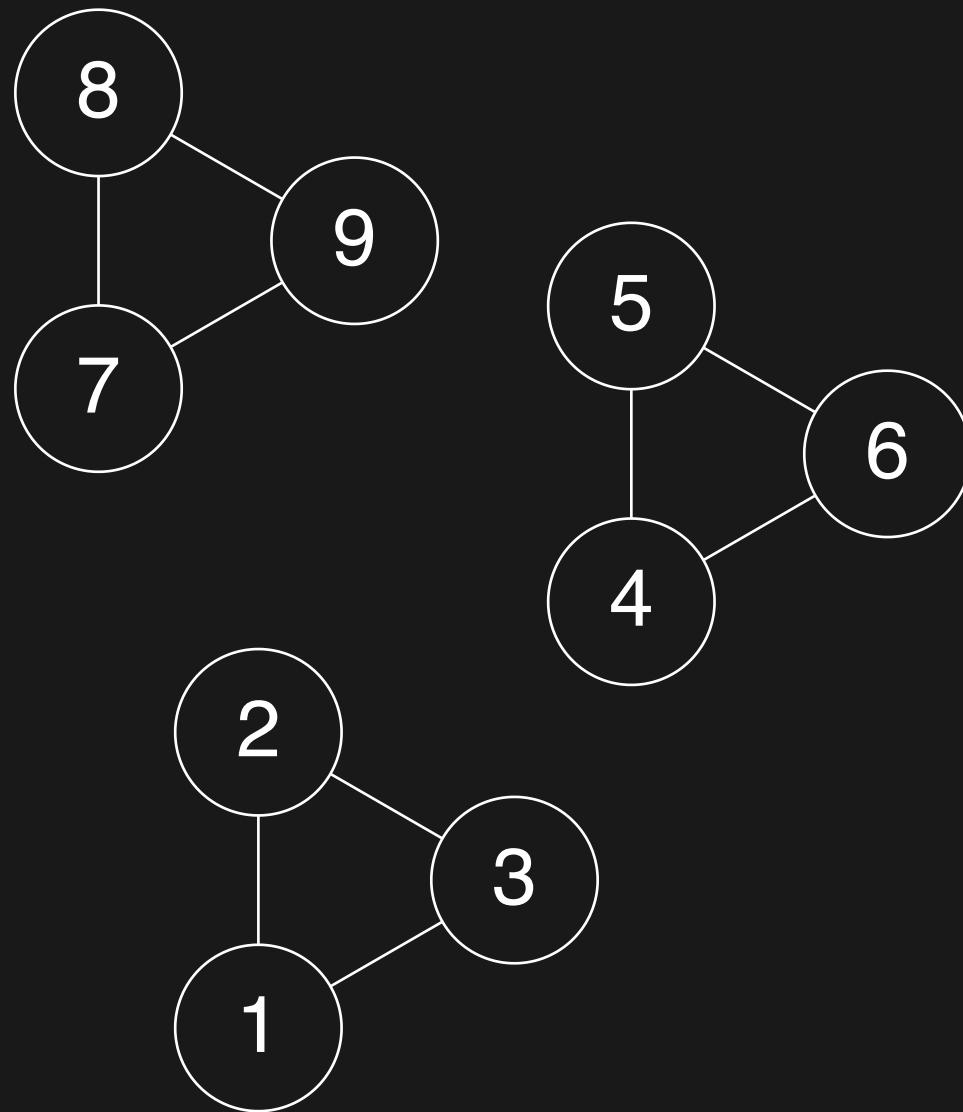
Entropy always increases

Second law of thermodynamics.

Thought experiment

-
- A diagram consisting of nine numbered circles arranged in a roughly triangular pattern. The circles are white with black outlines, set against a solid black background. The numbers are large and bold.
- Circle 1 is located in the bottom right corner.
 - Circle 2 is positioned below and to the left of Circle 1.
 - Circle 3 is located below and to the left of Circle 2.
 - Circle 4 is located above and to the left of Circle 3.
 - Circle 5 is located above and to the left of Circle 4.
 - Circle 6 is located above and to the right of Circle 4.
 - Circle 7 is located above and to the right of Circle 6.
 - Circle 8 is located below and to the right of Circle 7.
 - Circle 9 is located below and to the right of Circle 5.





Complexity

- Same number of nodes
- Edges: $\frac{n(n-1)}{2}$
 - Monolith: 36
 - Modular Monolith: $3 \times 3 = 9$

Complexity

$$36/9 = 4 \times$$

You're in good company.

<https://shopify.engineering/deconstructing-monolith-designing-software-maximizes-developer-productivity>

https://medium.com/@dan_manges/the-modular-monolith-rails-architecture-fb1023826fc4

Rails Engine

```
rails plugin new ./engines/core --full  
rails plugin new ./engines/api --mountable
```

`spec/dummy` is your friend

don't do this

```
# Source: https://tanzu.vmware.com/content/blog/leave-your-migration-to-a-database
initializer :append_migrations do |app|
  unless app.root.to_s.match root.to_s
    config.paths['db/migrate'].expanded.each do |path|
      app.config.paths['db/migrate'] << path
    end
  end
end
```

```
Rails.application.routes.draw do
  mount EDI::Engine => '/edi'
  mount API::Engine => '/api'
  mount Shop::Engine => '/'
end
```

Webpacker

Ask @coorasse

Alpha preview: Modern JavaScript in Rails 7 without Webpack

23,332 views • Aug 16, 2021

1.1K

21

SHARE

SAVE

...



David Heinemeier Hansson

15.6K subscribers

SUBSCRIBED



Modern web apps without JavaScript bundling or transpiling:
<https://world.hey.com/dhh/modern-web-...>

SHOW MORE

106 Comments

SORT BY



Add a public comment...



Bjorn Forsberg 1 month ago

One of the beautiful things about this approach is that it works REALLY well with rails engines and gem assets.. which has been a nightmare (near impossible) with Webpack 🙌

38 REPLY

Andrew Havens 1 month ago (edited)

Very cool. Webpack has been a pain in my side. Looking forward to moving to something simpler, especially in regards to Rails engines/gems providing their own javascript.

13 REPLY

JavaScript and Rails

world.hey.com



DAVID HEINEMEIER HANSSON

September 6, 2021

Rails 7 will have three great answers to JavaScript in 2021+

Rails has been unapologetically full stack since the beginning. We've continuously sought to include ever-more default answers to all the major infrastructure questions posed by modern web development. From talking to a database, to sending and receiving emails, to connecting web sockets, to rendering HTML, to integrating with JavaScript. This full-stack strategy has been key to the success of Rails, but it also remains an enduring source of controversy. What's too much to include? What's not enough?

To consistently answer that evergreen question, we look to [The Rails Doctrine](#), and especially the third pillar of [The Menu Is Omakase](#). It's why we fret so much over the defaults, but also why the option to substitute is so crucial.

There's been no more fretting over the defaults, or a closer examination of the substitutes, than with the JavaScript part of the question over the years. And especially

<https://world.hey.com/dhh/rails-7-will-have-three-great-answers-to-javascript-in-2021-8d68191b>

Three choices for JS



1. import maps
2. jsbundling-rails
3. Rails API with separate SPA

The 4th choice

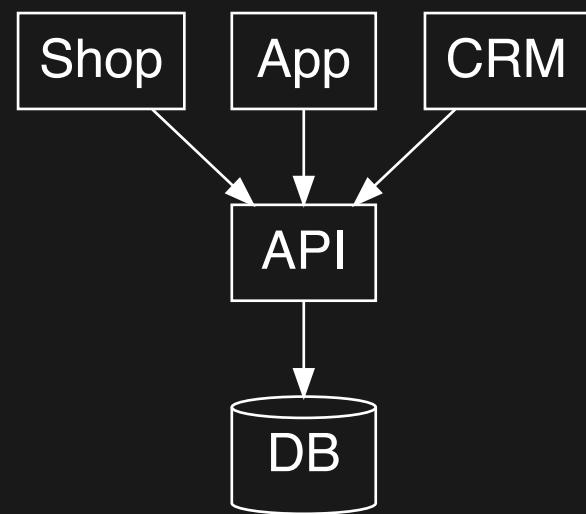
Simple. Obvious. Best of both worlds?

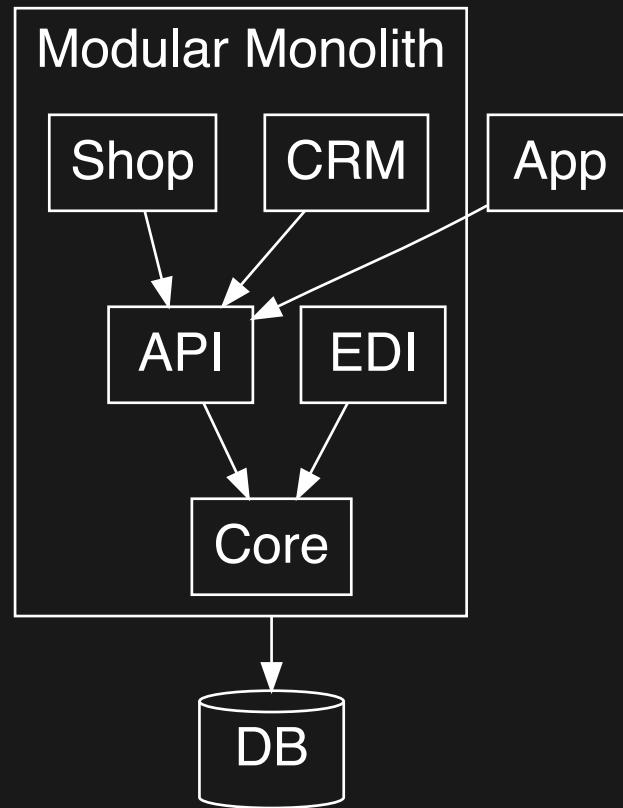


- 1 repo
- 1 CI
- 1 deployment (Heroku)
- different stacks
- clients/crm
- public/crm → ../../clients/crm/dist

Works with Rails 0.5

RSpec and FactoryBot.
End-to-end testing.





Momo

Modular Monolith

Hoelzle API 0.1.0 OAS3

Find products, create orders and manage billings.

[Hoelzle AG - Website](#)

[Authorize](#)



Filter by tag

shop Endpoints used by website and shop.



GET /products



GET /products/{id}



GET /products/{product_id}/neighbours



GET /categories



GET /categories/{id}

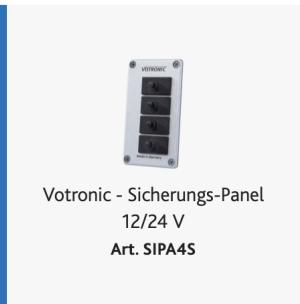


GET /orders





Votronic - Schalter-Panel
12 V
Art. SchaPa4S



Votronic - Sicherungs-Panel
12/24 V
Art. SIPA4S



ABS/EBS-Anbausatz für
Zugfahrzeuge mit 12 V
Art. ABS/EBS/Set



DEFA Kabletrommel 5
Meter, 2.5 mm²
Art. 499900



Vorschaltgerät für
Tagfahrlicht zu 80AW12 und
80AW24
Art. DRLHOTDOG



EcoFlow Power Station
RIVER PRO, 720 Wh
Art. EFRiver/PRO



Halteplatte zu
Sicherungshalter ATO In-Line
Art. H1307



Einbausteckdose mit flacher
Montageplatte - 1 x USB-A
und 1 x USB-C
Art. 50191

Shop 

Wissen



Solarenergie: Mit einem Camper
unbeschwert die Ferien geniessen



AC-Lademöglichkeiten für E-Autos

ABC der E-Mobilität



Logistik und die aktuellen
Herausforderungen



Die Ausphasung ineffizienter
Leuchtmittel geht weiter

Hoelzle AG

CRM Login

Benutzername
E-Mail-Adresse angeben...

Passwort
Passwort eingeben...



Anmelden

There is no right or wrong.
Be pragmatic.

Questions?

Thanks

TODO

- Core engine
- API engine
- JS / Svelte client
- Administrate
- Shop
- Device
- Github pages
- Omniauth
- Standardrb
- RSpec
- CI

- ■ Ruby 3 / Rails 7
- ■ Hotwire
- ■ Importmaps
- ■ Spyke
- ■ ActionCable
- ■ Docker
- ■ install:migrations NOT

Why API?

1. Separation of concerns
2. Different kinds of clients
3. Offline / native clients
4. Technological independence / decoupling
5. Standards such as JsonAPI
6. Every developer knows what it is
7. Minimal attack surface
8. Future proof (Jamstack and what not)