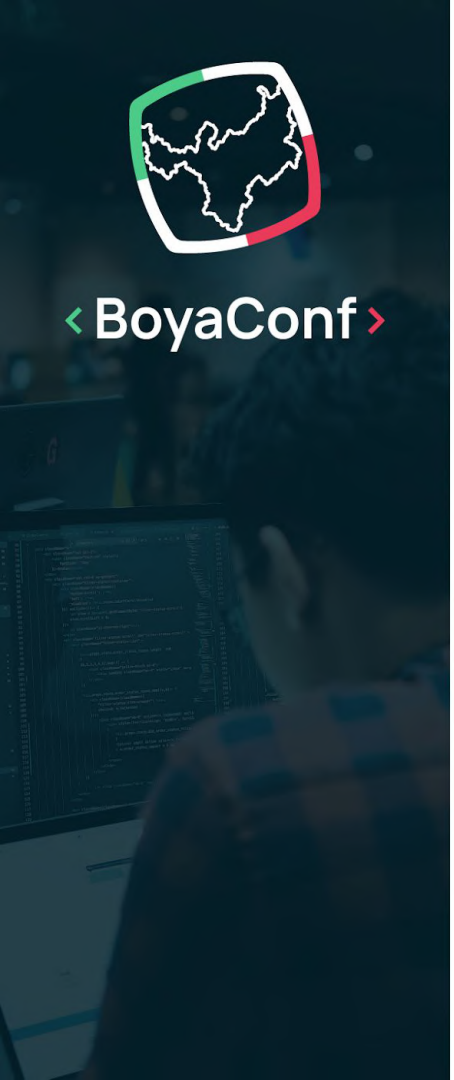
Domain Driven Design applied to Frontend

Milena Suarez

2019-11-09



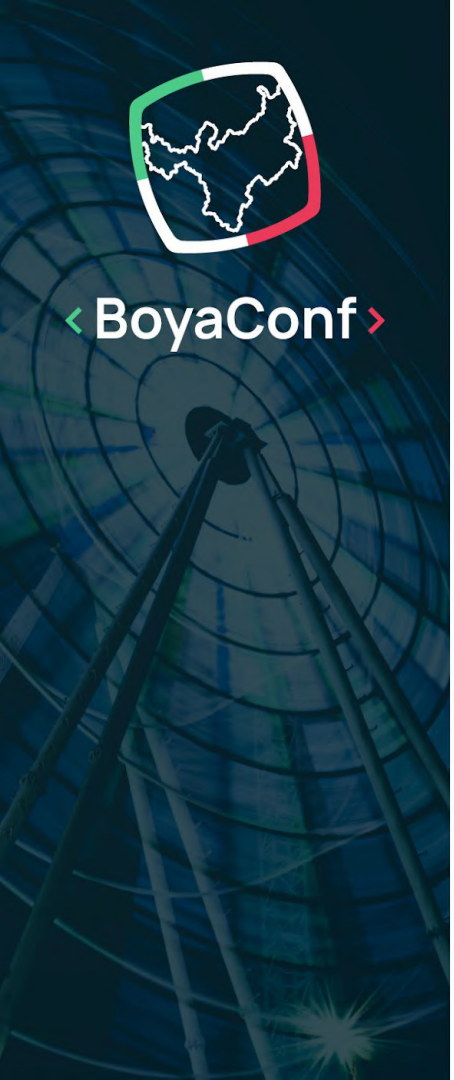
Milena Suarez

Senior Frontend engineer. I have worked in academic, retail and financial industries with Javascript, Typescript, Angular.

I love the console, travels, wine and cats.

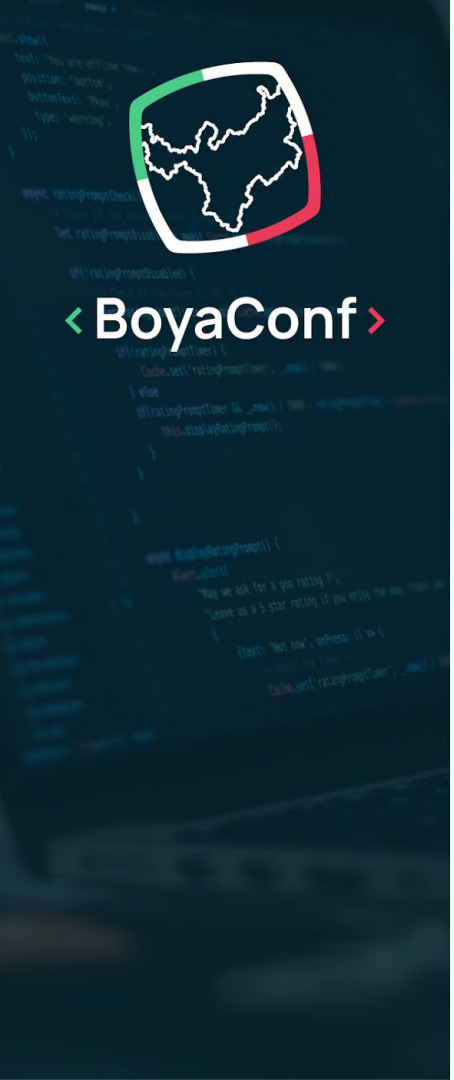
@milena_suarezl

@milenasuarezl



Agenda

1. What is DDD?
2. Why DDD?
3. How to do DDD?
4. Strategic Design
5. Tactical Design
6. Case Study
7. Advantages
8. Caveats



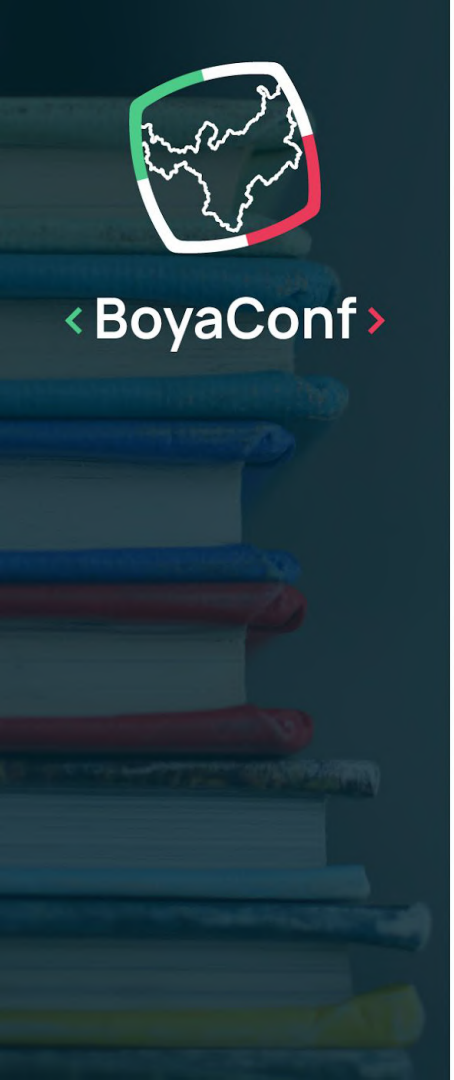
What is Domain driven design?

DDD is a **software development approach**.

"DDD reminds you of common-sense rules, and it offers a set of ideas, principles, and patterns to make it happen."

Vaughn Vernon

Domain-Driven Design Tackling Complexity in the Heart of Software. 2003. Eric Evans

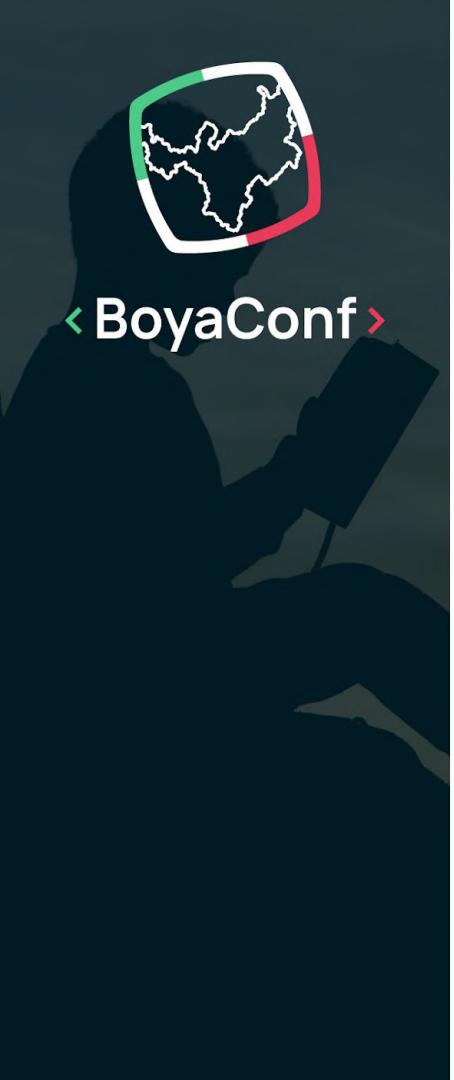


Why DDD?

"The economy of no design is a fallacy"

Vaughn Vernon

- Effective design
- Improve your craft
- increase your success in projects
- Model unique business needs
- Help your business compete at new heights

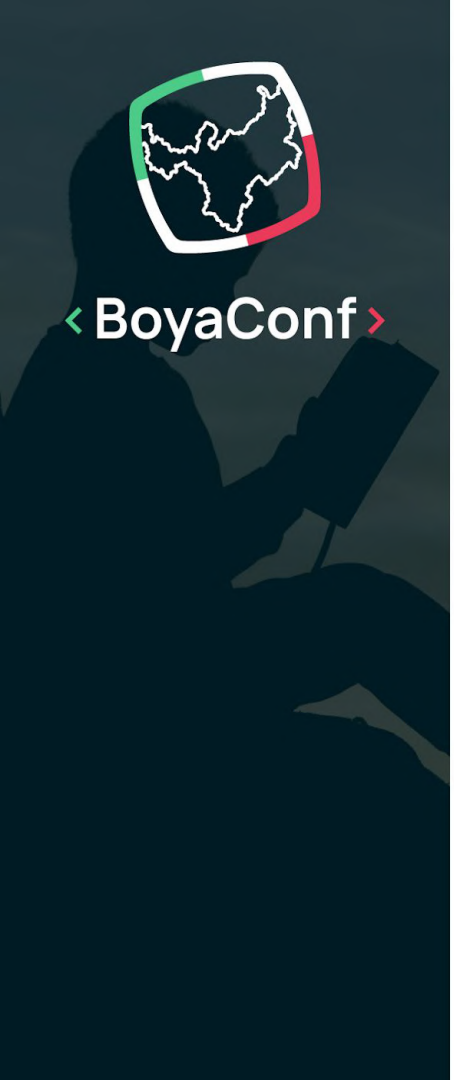


How to do DDD?

“Embrace to business people; They’ll enjoy explaining their jobs and problems, you’ll love listening and finding solutions.”

"In order to go fast, we must go well."

khalil stemmler

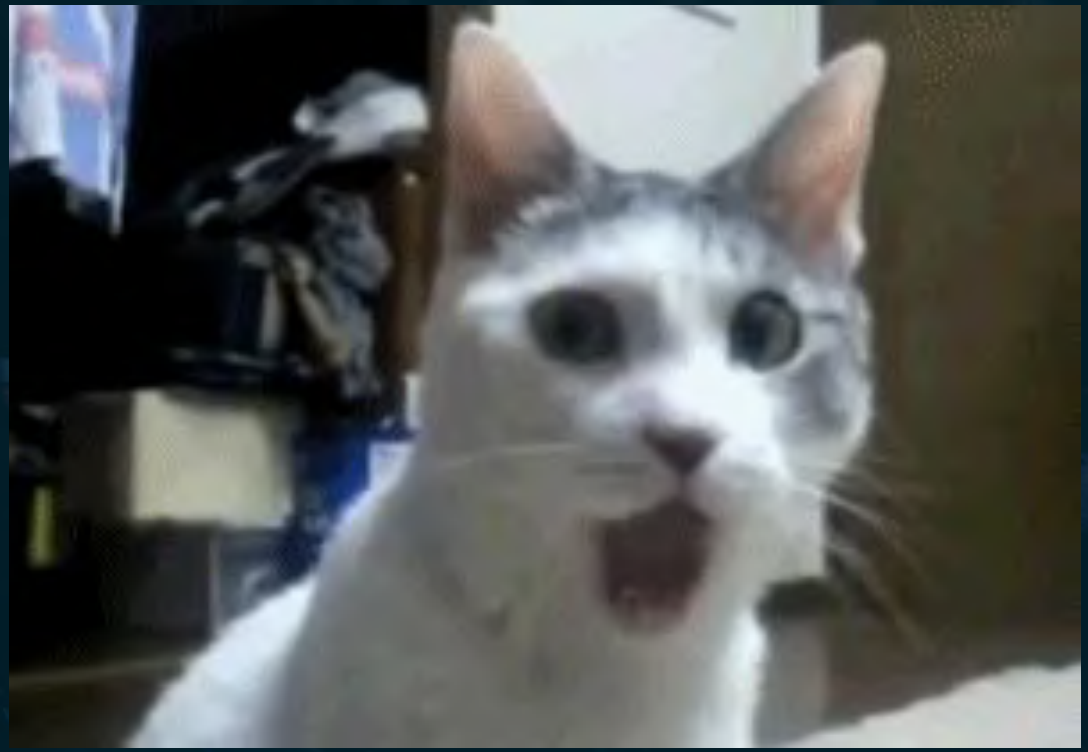


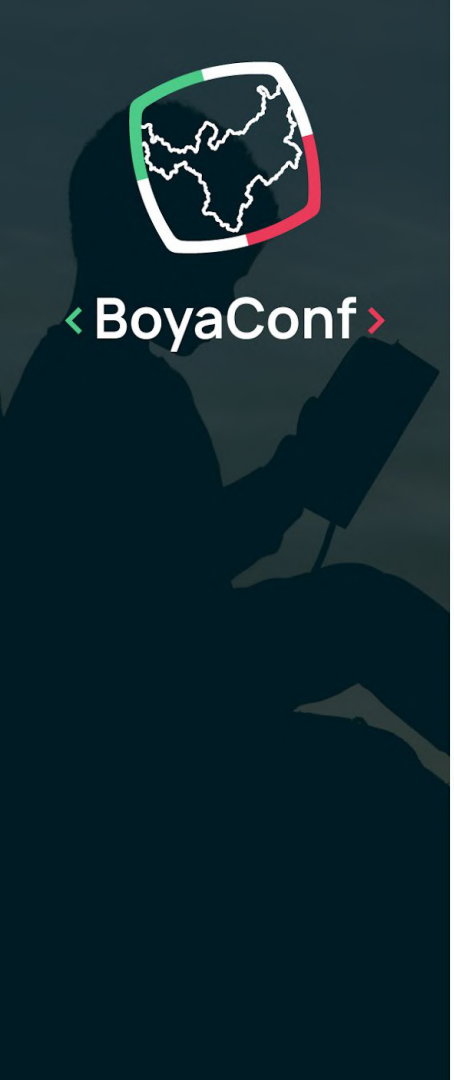
Strategic Design

- Ubiquitous Language
- Domain
- Domain Model
- Sub-Domain
- Bounded Context
- Core-Domain
- Context
- Generic-Domain
- Context Map

Tactical patterns

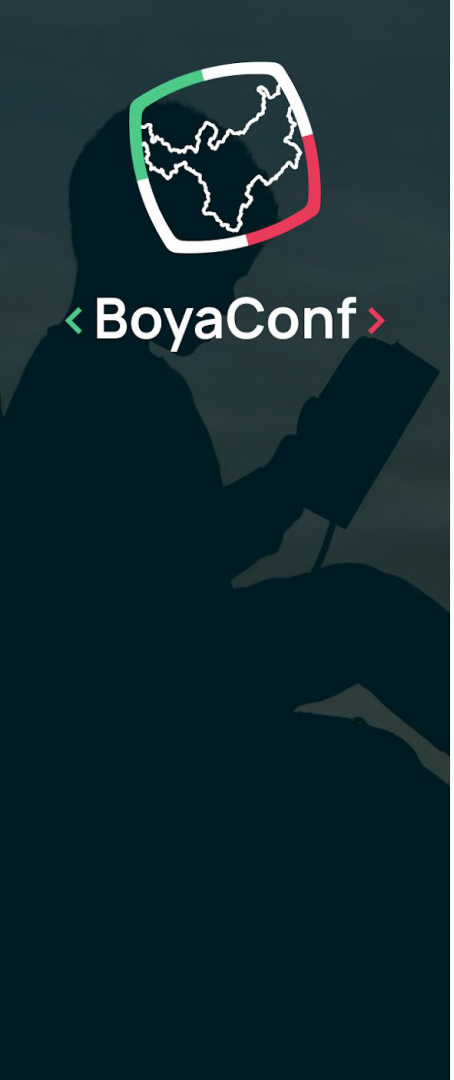
- Entities
- Layers
- Value Objects
- Modules
- Dependency Injection
- Services
- Factories
- Repositories
- Aggregates





Strategic Design

- Identify the core competencies
- Introduction to new staff
- Ubiquitous language



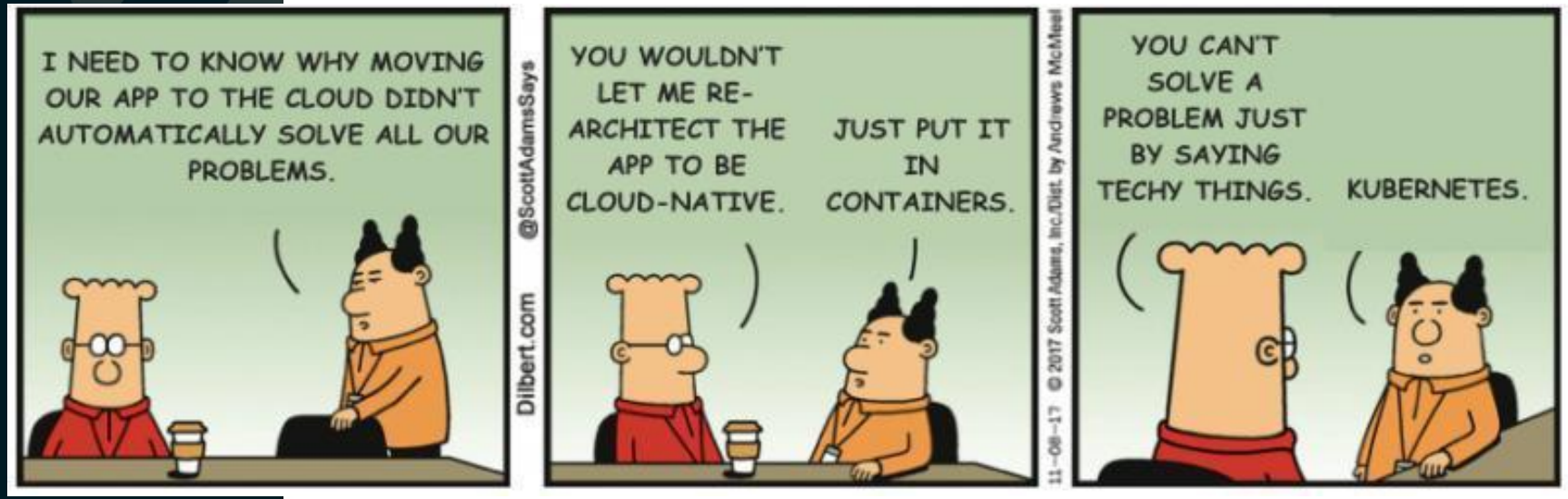
Ubiquitous language

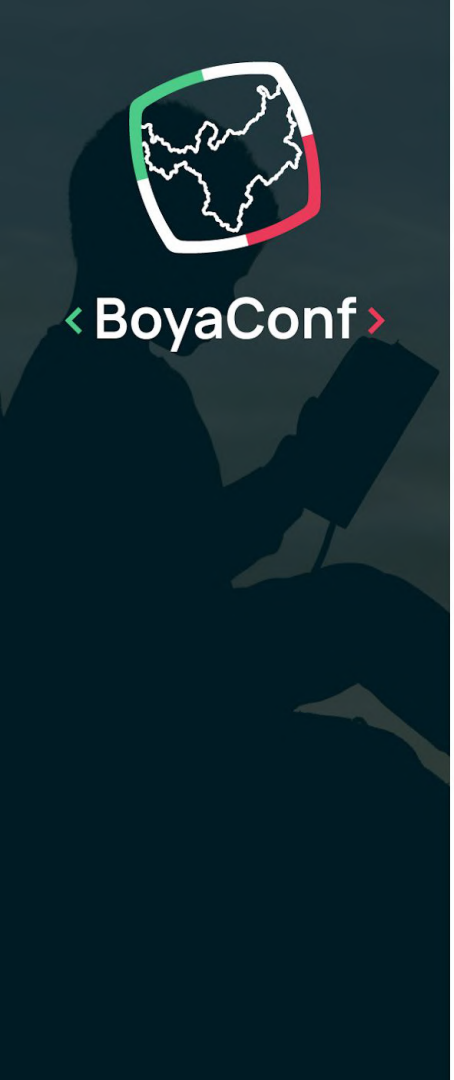
“The names should be agreed on with the business.”

Why? Because in the future, when you talk about a new feature or a bug report the domains expert understand you"

Think and write meaningful names.

Communication between domain experts and developers.





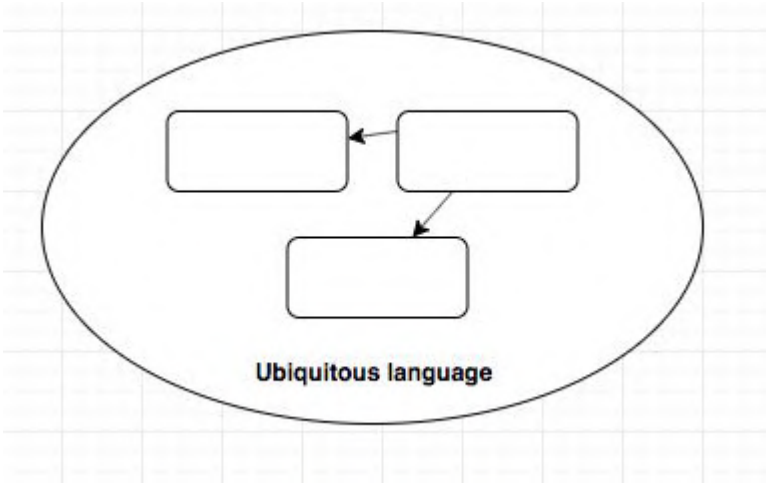
Bounded Context

Define the context for business people.

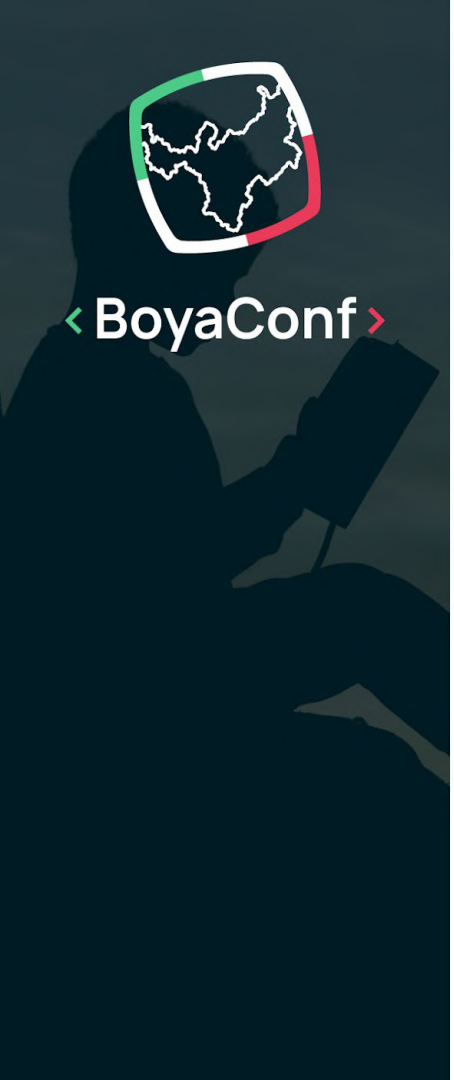
Why is the Core?

Subdomains

- Core Domain
- Generic Domain

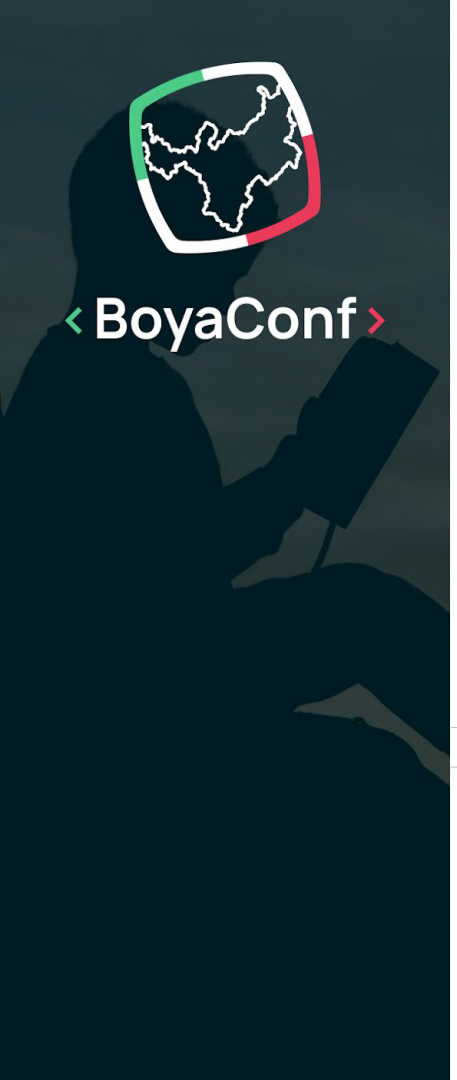


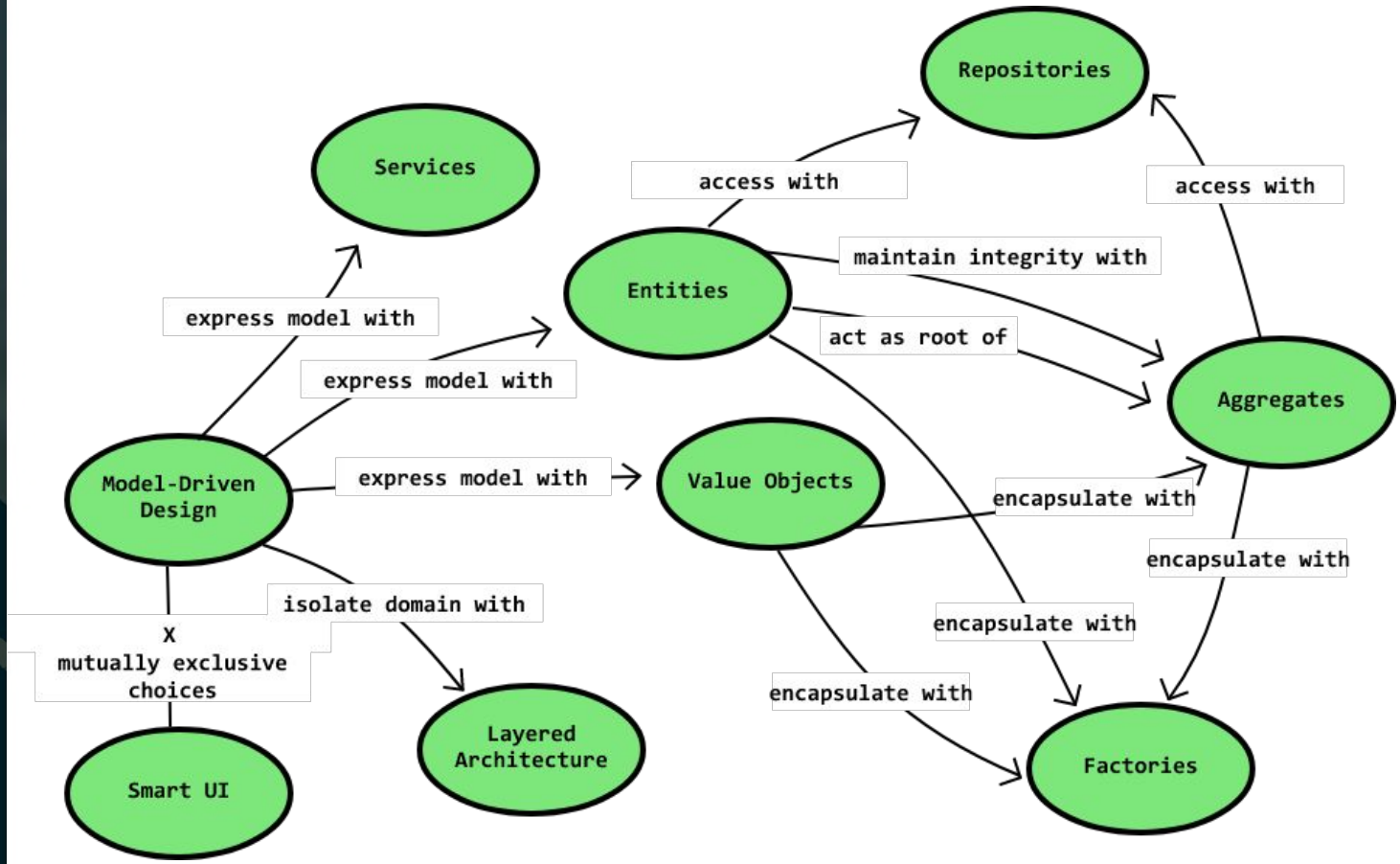
Context Mapping



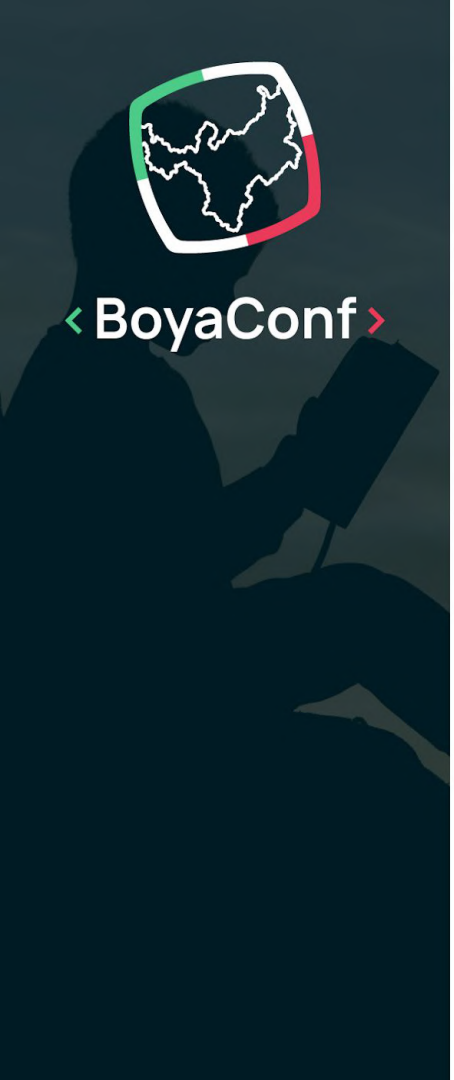
Tactical design

- Applying the tactical patterns
- Protecting the domain layer
- Anemic Domain Model

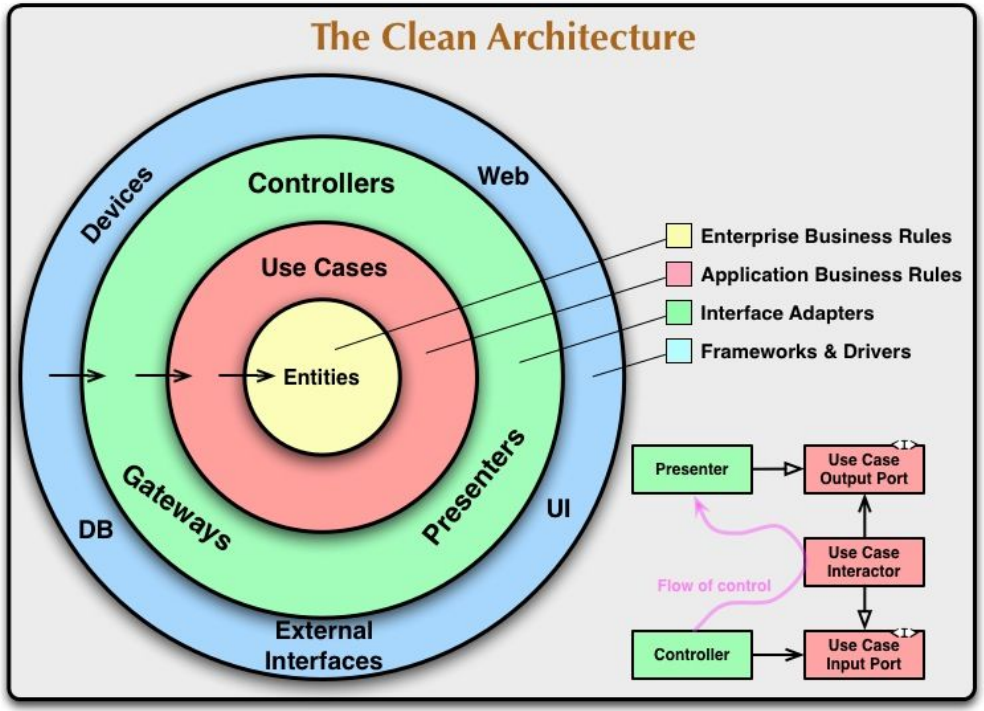




Taken from khalilstemmler.com



Protecting Domain Layer - Hexagonal architecture

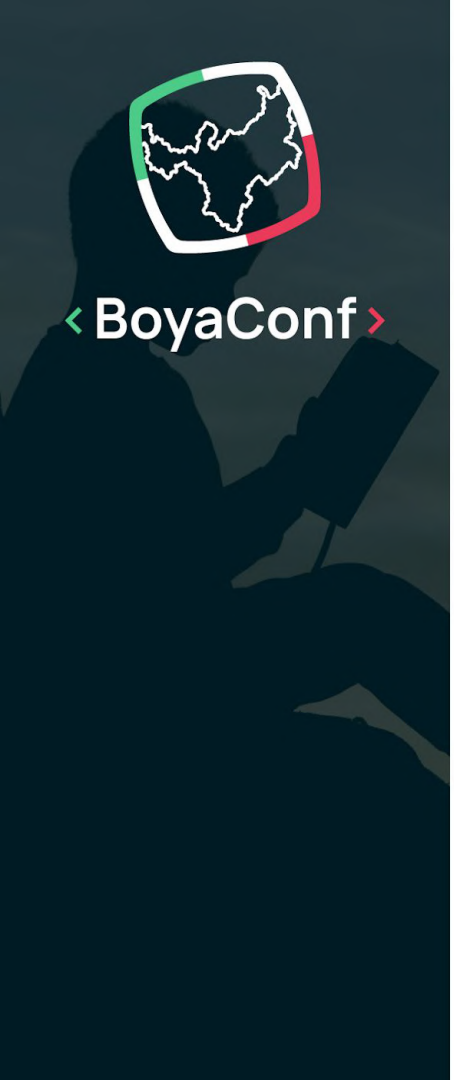


https://www.slideshare.net/hanamirb_sp/clean-architecture-1-hanami-sao-paulo-meetup

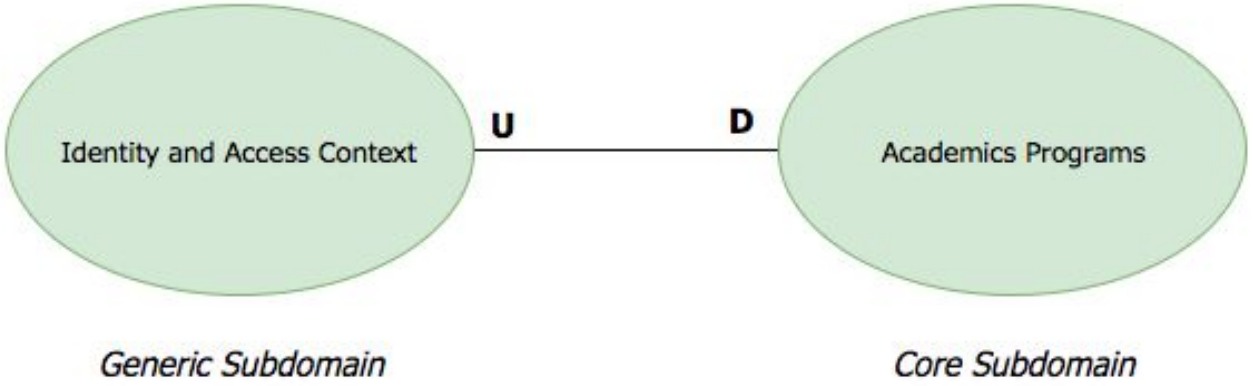
Anemic Domain Model

Employee
+ identification: string + name: string + hoursWorked: number
+ save(identification) + updated(identification, data)

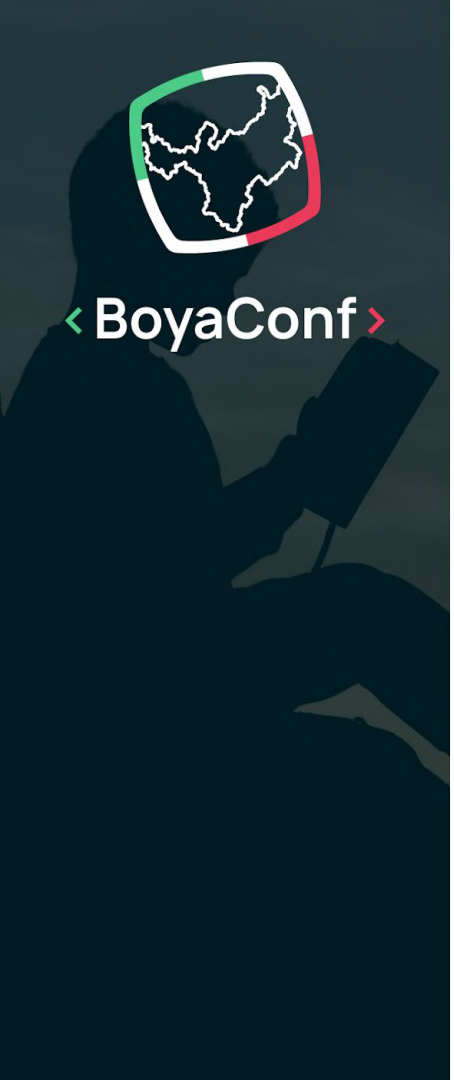
Employee
+ identification: string + name: string + hoursWorked: number
+ save(identification) + updated(identification, data) + calculatePay(identification) + calculateHoliday(identification)

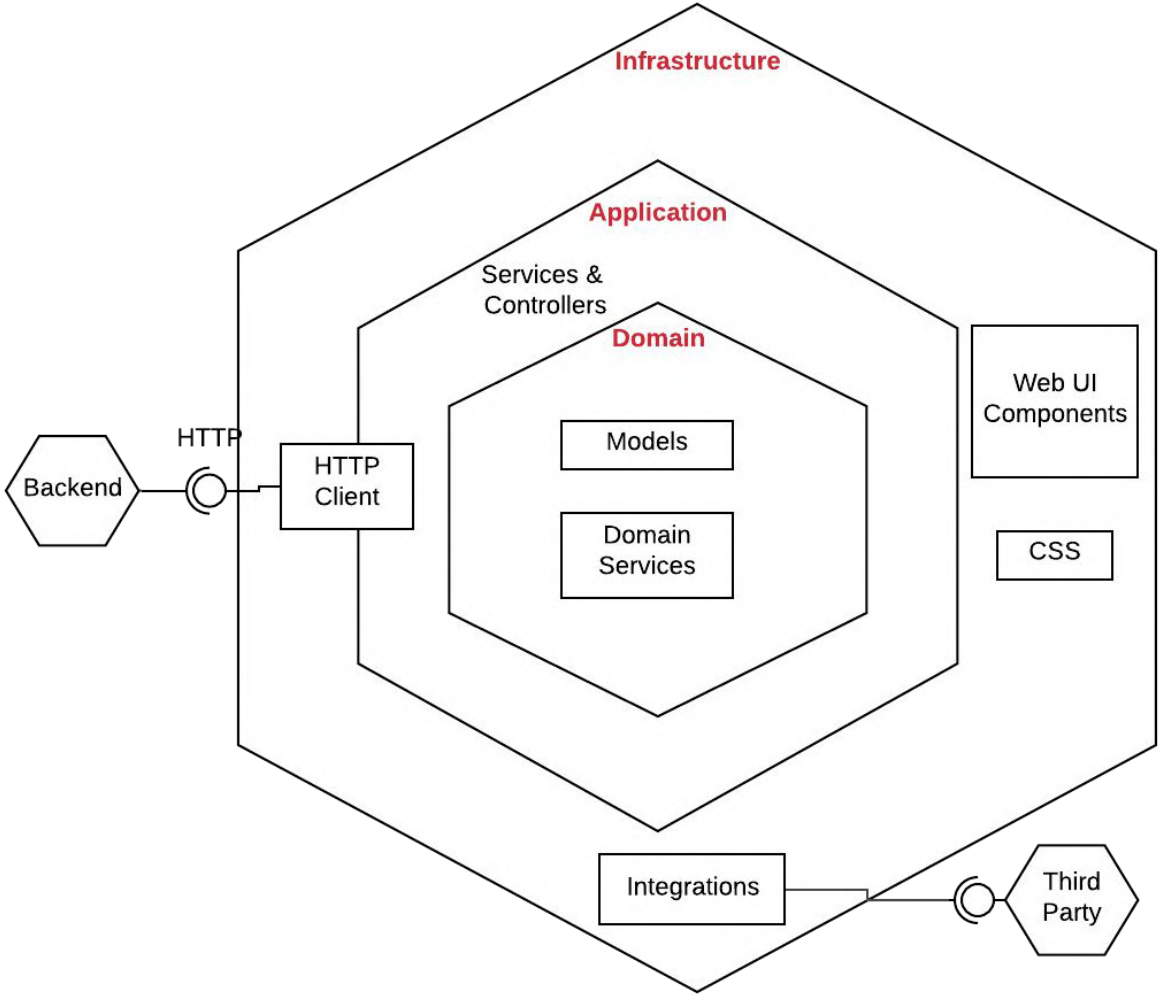


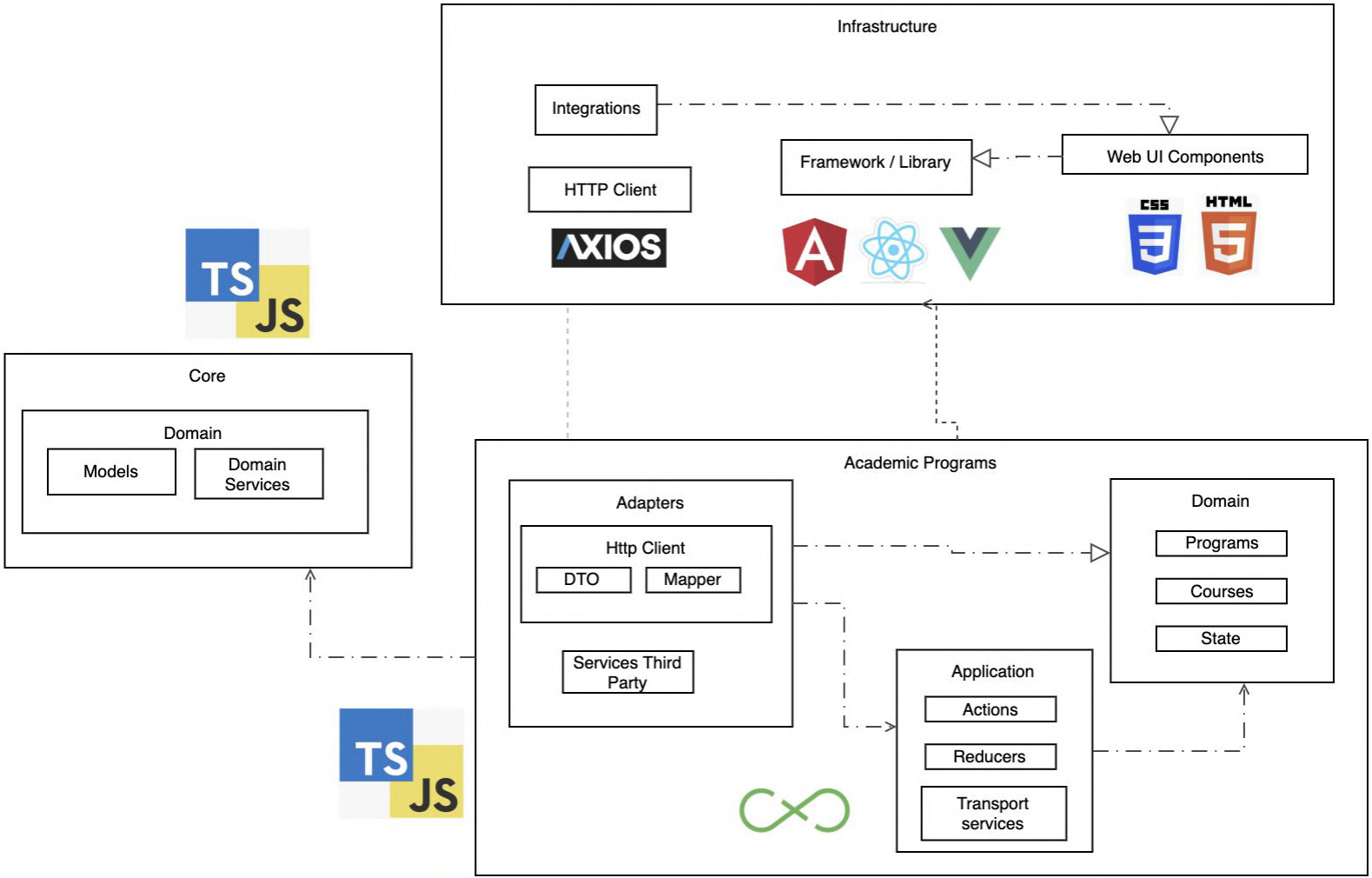
Case Study



U	Upstream	Service exhibitor
D	DownStream	Service consumer

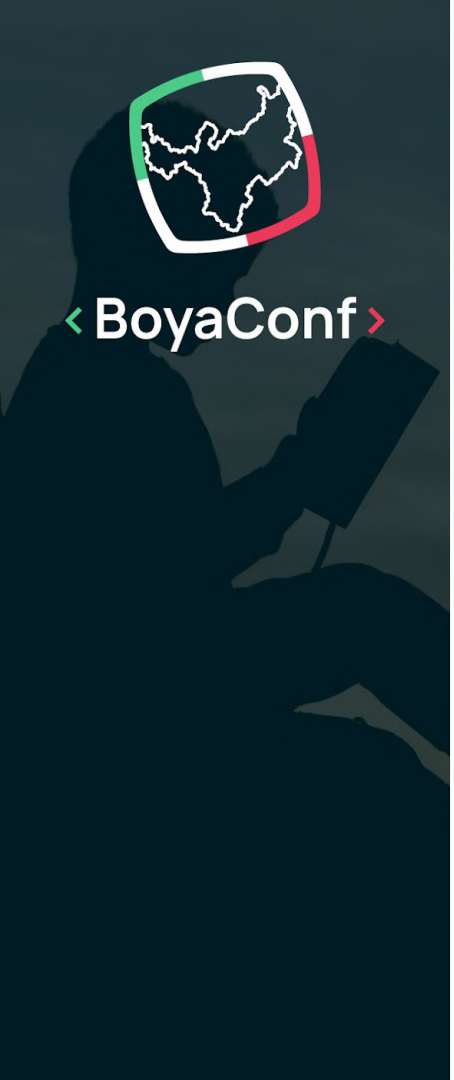

BoyaConf


Infrastructure
Application
Services &
Controllers
Domain
Models
Domain
Services
Web UI
Components
CSS
HTTP
Backend
HTTP
Client
Integrations
Third
Party


Infrastructure
Integrations
Framework / Library
Web UI Components
HTTP Client
AXIOS
CSS
HTML
TS
JS
Core
Domain
Models
Domain Services
Academic Programs
Adapters
Http Client
DTO
Mapper
Services Third Party
Domain
Programs
Courses
State
Application
Actions
Reducers
Transport services
TS
JS

Advantages

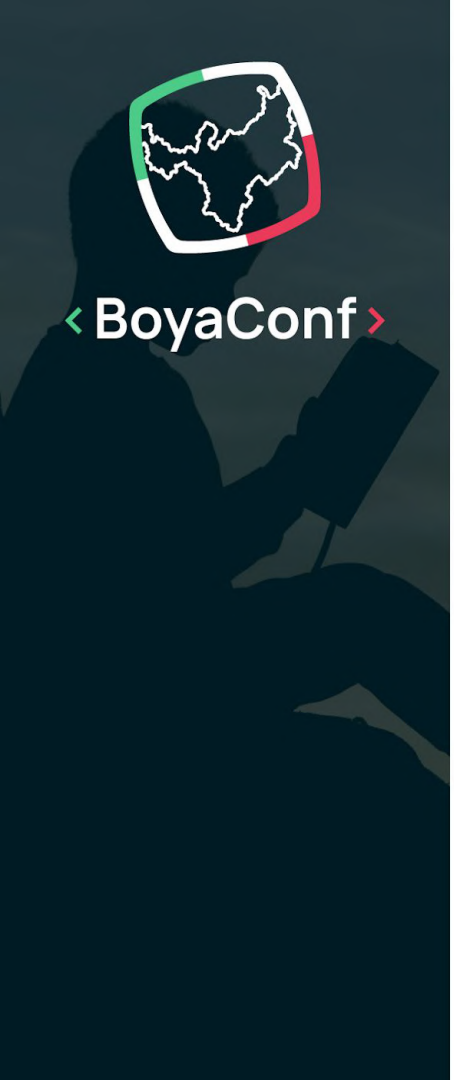
- **A better understanding** of business by domains experts and developers
- Business logic **agnostic** to the development tool
- Controlled team **growth**
- Large scale applications a lot more **friendly** and **maintainable**
- Write meaningful unit **tests** that covered **the *business rules***



Caveats

> *"The software development is considered a cost center rather than a profit center"*
>
> *Vaughn Vernon*

- Acquired **knowledge** of all members
- It is an **iterative** approach
- Required effort between **team members**



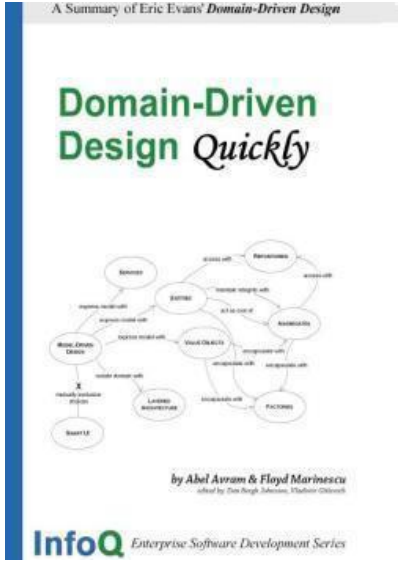
Conclusion

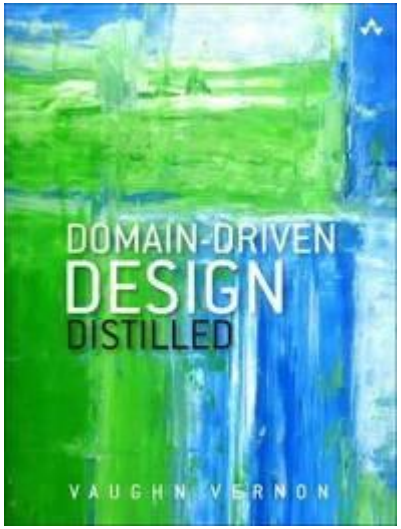
Domain Driven Design allows ***being better developers*** *according to needs the business*

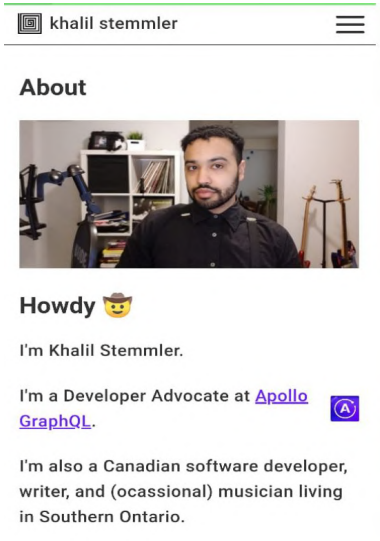
Grow into the company

The frontend developments is not only html and css, also it is really important a effective design before to write code.

Interests resources








THANK YOU!
EVERYDAYIDERP.TUMBLR.COM